Australian Government
Department of Defence
Defence Science and
Technology Group

Linear-Elastic 2D and 3D Finite Element Contact Analysis of a Hole Containing a Circular Insert in a Fatigue Test Coupon

Witold Waldman

Aerospace Division
Defence Science and Technology Group

DST-Group-TR-3134

ABSTRACT

Aircraft structures typically contain large numbers of circular holes that are fitted with fasteners such as bolts or rivets. During the service life of aircraft, fatigue damage often occurs at such holes. The accurate analysis of stress distributions occurring around the boundary of holes in the presence of fasteners is therefore an important consideration during studies of fatigue life and test interpretation activities supporting full-scale fatigue test programs. In the present work, two-dimensional linear-elastic plane elasticity solutions for contact stresses caused by a circular disk inserted into a circular hole in an infinite plate undergoing remote loading have been implemented in a FORTRAN program. These were used to validate the contact stress distributions for a circular hole in an aluminium plate fitted with a titanium fastener that were computed using two-dimensional finite element contact analysis. By application of a finite-width correction factor, the analytical infinite-plate solutions were also used as a point of comparison with the results produced by subsequent two-dimensional and three-dimensional finite element contact analyses of a finite-width fatigue test coupon. The results obtained here are useful for aircraft structural integrity analysis work, and subsequent analyses of contact problems such as this one can be expected to be accurate so long as sufficiently refined finite element meshes are utilised.

RELEASE LIMITATION

Approved for public release

Published by

Aerospace Division
Defence Science and Technology Group
506 Lorimer St
Fishermans Bend, Victoria 3207, Australia

Telephone: 1300 333 362
Fax: (03) 9626 7999

AR-016-357
July 2015

APPROVED FOR PUBLIC RELEASE

Linear-Elastic 2D and 3D Finite Element Contact Analysis of a Hole Containing a Circular Insert in a Fatigue Test Coupon

Executive Summary

Aerospace Division is presently comprehensively involved in developing and applying technologies that ensure the safety and enhance the availability of aircraft in service with the Royal Australian Air Force. Many of these aircraft structures typically contain large numbers of circular holes that are fitted with fasteners. Fatigue damage often occurs at such holes during the service life of aircraft. The accurate analysis of stress distributions occurring around the boundaries of holes in the presence of fasteners is therefore a very important consideration in fatigue life studies, many of which involve extensive and expensive experimental fatigue testing of structurally-representative coupons undergoing sequences of programmed loading.

Prior two-dimensional linear-elastic plane elasticity solutions are available for computing contact stresses caused by a circular disk inserted into a circular hole in an infinite plate undergoing remote loading. One of the known available solutions is applicable to the commonly-occurring case where the plate material and the insert material have different elastic properties, which is relevant to the situation that typically occurs in aircraft structures. However, as originally formulated, the solutions do not take into account any finite-width or three-dimensional effects, both of which have an important influence on the stress distribution and hence the resulting fatigue life of the structure.

In the present work, these linear-elastic analytical/numerical hole–insert contact solutions have been implemented in a custom-written FORTRAN program. They were then used to validate the contact stress distributions associated with a circular hole in an aluminium plate fitted with a titanium fastener that were computed using two-dimensional finite element contact analysis. By application of a finite-width correction factor, the infinite-plate solutions were also used as a point of comparison with the results produced by subsequent two-dimensional and three-dimensional finite element contact analyses of an actual fatigue test coupon. The results obtained from the present three-dimensional analysis of the fatigue test coupon provide improved stress distribution data for use in the validation of test interpretation activities relating to full-scale fatigue testing of aircraft structures in service with the RAAF. These three-dimensional contact stress solutions have been extensively validated and are useful in aircraft structural integrity analysis work, and the methodology described here provides some guidance as to how to perform contact analysis with high accuracy. Any subsequent contact analyses of other configurations can be expected to be reliable and accurate so long as suitably refined finite element meshes are utilised.

Author

Witold Waldman
Aerospace Division

Mr Witold Waldman completed a BEng (with distinction) in Aeronautical Engineering at the Royal Melbourne Institute of Technology in 1981. He commenced work in Structures Division in 1982 at what was then the Aeronautical Research Laboratory. He has published a number of papers and reports, and his experience has focussed on stress analysis using finite element and boundary element methods, structural mechanics, fracture mechanics, computational unsteady aerodynamics, structural dynamics testing, digital filtering of flight test data, nonlinear optimisation, and spectral analysis. His recent work has been in the areas of structural shape optimisation and computation of stress intensity factors. He is currently a Senior Research Engineer in the Structural and Damage Mechanics Group in the Airframe Technology and Safety Branch of Aerospace Division within the Defence Science and Technology Group, Department of Defence.

Contents

Nomenclature

D	diameter of circular hole in plate
E	Young's modulus
$E(k)$	complete elliptic integral of the second kind
H	height of plate
k	modulus of complete elliptic integral
$K(k)$	complete elliptic integral of the first kind
K_t	stress concentration factor
K_{tg}	gross-section stress concentration factor
r	distance away from stress singularity
S	remote stress
S_x	remote stress aligned with x-direction
S_y	remote stress aligned with y-direction
t	thickness of plate
W	width of plate
x	rectangular Cartesian x-coordinate
y	rectangular Cartesian y-coordinate
z	rectangular Cartesian z-coordinate
θ	polar angle
η	contact angle between plate and insert
υ	Poisson's ratio
σ_r	radial stress
σ_t	tangential stress
∞	infinity

1. Introduction

Aerospace Division designs and uses metallic fatigue testing coupons to aid in the fatigue life and aircraft structural integrity management of RAAF airframes. An aluminium coupon has been previously designed in support of the independent verification and validation of test results associated with the Lead-In Fighter (LIF) Hawk full-scale fatigue test being undertaken by BAE Systems for the RAAF. This coupon is known as the LIF Hawk Filled Hole Coupon (see Figure 1), and it is intended to be used in fatigue crack initiation and crack growth studies.

The fatigue test coupon has been the subject of a previous detailed finite element analysis (FEA) to provide engineering data on the behaviour of the stress concentration factor (SCF), K_t, as a function of both the applied tension-only load and the diametral gap between the hole and a close-fit titanium fastener that was inserted in the hole. A reduction in K_t occurs because, upon contact of the hole edge with the fastener, load transfer between the fastener and the plate will occur in the transverse direction under a tensile load, thus propping open the hole. This report covers the work that has subsequently been carried out to validate the use of finite element analysis in analysing both two-dimensional (2D) and three-dimensional (3D) contact problems.

When a fastener is inserted into the hole in the fatigue test coupon, prior work has indicated that the greatest reduction in K_t is obtained for the case of a neat-fit insert (i.e. one with a diametral gap of zero). Brombolich [1] noted that it has been experimentally verified that the fatigue life can be improved when close-tolerance fasteners are installed in holes. Hence, the present FEA study focusses on the neat-fit case, which serves to provide a lower bound for the reduced value of K_t, in conjunction with the upper bound provided by the open-hole case. The neat-fit case is an example of a conforming contact problem, as the boundary of the hole and the surface of the insert touch at multiple points before any deformation occurs.

For 2D frictionless contact problems involving a smooth circular disk inserted into a circular hole in an infinite plate loaded at infinity, two sets of analytical solutions are known to be available. The first of these was obtained by Stippes, Wilson and Krull [2] for a plate being uniaxially-loaded in tension, where the plate and the disk have the same elastic material properties. The second solution was obtained by Wilson [3] and went beyond this, dealing with biaxial loading and a disk that had different elastic material properties than the plate. Both of these solutions provide valuable analytical results that have been utilised here as benchmarks for direct comparison with the FEA-based contact analysis solutions.

Section 2 provides details of the geometry of the LIF Hawk Filled Hole Coupon and the elastic material properties of the coupon and the fastener. An exposition of the two available analytical solutions that are relevant to the coupon–fastener contact analysis is presented in Section 3. The general approach that was utilised in performing the contact analyses using the Abaqus FEA code is presented in Section 4. The 2D finite element contact analyses of an analytical benchmark problem and the LIF Hawk Filled Hole Coupon are described in Sections 5 and 6. The 3D contact analysis using different levels of mesh refinement is covered in detail in Section 7. Subsequently, a discussion of the 2D and 3D analysis results is

presented in Section 8. A summary of the key results is provided in Section 9. Some general conclusions are offered in Section 10.

2. Geometry and material properties

The general geometry and dimensions of the LIF Hawk Filled Hole Coupon are as shown in Figure 1. The coupon has the common "dog bone" shape, with a 2:1 narrowing at the centre section relative to the width at the two ends. The coupon plate thickness is t = 6 mm, and a hole of nominal diameter d = 6.35 mm is located in the centre of each coupon. In the 2D analysis work that follows, the origin of the x-y coordinate system is located at the centre of the hole in the coupon, as shown in Figure 1. For the 3D analyses, the origin of the x-y-z coordinate system is located at the centre of the hole in the coupon at the midplane of the plate, with the z-direction coming out of the page according to the right-hand rule.

The coupon (plate) material is an aluminium alloy with a nominal yield strength of 405–450 MPa, Young's Modulus of E = 69.0 GPa, and Poisson's ratio of υ = 0.33. The fastener (pin, insert, bolt) is made from a titanium alloy with a nominal yield strength of 880 MPa, E = 113.8 GPa, and υ = 0.31.

The results of the present work are valid up to the elastic limit of the material. Any results beyond this, although not entirely accurate, may be useful as an estimate for the general behaviour of the hole–pin contact interface at higher loads.

3. 2D analytical solutions

Consider the general geometrical configuration of an idealised infinite 2D elastic plate loaded by stresses S_x and S_y at infinity, as depicted in Figure 2. The origin of the x-y coordinate system is located at the centre of the circular hole, which is filled with a conforming neat-fit elastic circular disk insert. The contact angle between the plate and the insert is η, and the angle θ is the rotation from the x-axis (positive in the anti-clockwise direction). The Young's modulus and Poisson's ratio of the elastic plate are E_p and υ_p, and for the elastic disk insert they are E_i and υ_i.

3.1 Plate and pin with identical elastic material properties

In their 1962 conference paper, Stippes, Wilson and Krull [2] provided a 2D analytical solution to the frictionless contact problem of a smooth circular disk inserted into a circular hole in an infinite plate loaded in uniaxial tension (S_x = 0, S_y > 0) at infinity. Their solution applies to the case where the plate and the insert have the same elastic material properties (E_p = E_i and υ_p = υ_i), and the initial diameter of the disk is the same as the hole. The applied uniaxial stress results in partial separation between the surfaces of the plate and the disk, and under linear-elastic conditions the results are independent of load level.

The contact angle η is obtained by iteratively finding the smallest root of a nonlinear equation involving trigonometric and logarithmic terms (using a method proposed by Shampine and Watts [4]), as well as the complete elliptic integrals of the first and second

kind, $K(k)$ and $E(k)$ (computed using a method proposed by Thatcher [5]), with modulus k = cos(η). The equation to be solved is:

$$4\sin^2\eta\left(2+\ln(\cos\eta)\right)K(k)-\left(2+\sin^2\eta+2(1+\sin^2\eta)\ln(\cos\eta)\right)E(k)=0 \qquad (1)$$

Stippes, Wilson and Krull [2] evaluated the extent of the contact arc to be η = 19.62°, while in the present work it has been calculated to be η = 19.62506°, which is in excellent agreement.

The equations for computing the radial stress along the contact arc, σ_r, as well as the tangential (circumferential) stress around the boundary of the hole, σ_t, were obtained to be:

$$\sigma_r(\theta)=\frac{3S_y}{2}\cos\theta\sqrt{\cos^2\theta-\cos^2\eta}-\frac{A}{2}\ln\left[\frac{\cos\theta+\sqrt{\cos^2\theta-\cos^2\eta}}{\cos\eta}\right], \quad \begin{matrix}-\eta\le\theta\le+\eta \\ \pi-\eta\le\theta\le\pi+\eta\end{matrix}$$
$$=0 \text{ elsewhere} \qquad (2)$$

$$\sigma_t(\theta)=S_y(1+2\cos 2\theta)+4B-\sigma_r(\theta)$$

where

$$A=-\frac{S_y\left(2+3\sin^2\eta\right)}{2\left(2+\ln(\cos\eta)\right)}$$
$$B=-\frac{1}{2}\left(A+\frac{S_y}{2}\right) \qquad (3)$$

Note that the radial stress σ_r is non-zero only along the length of the contact arc, being zero elsewhere. In the equation for σ_t, the first term corresponds to the well-known classical equation for the tangential stress around a circular hole in an infinite plate loaded uniaxially in tension [6], which produces a K_t of exactly 3 at θ = 0° and 180°, and a K_t of –1 at θ = ±90°.

Being relatively compact and simple, the above expressions are amenable to being used in spreadsheet calculations, and in fact they have been successfully programmed as functions in Microsoft Excel (see Appendix A). Plots of the normalised radial and tangential stresses, σ_r/S_y and σ_t/S_y, around the boundary of the hole are shown in Figure 3. The tangential stress for an empty circular hole is also shown there for comparison purposes, and the similarity to the results for the plate with an insert is clearly evident. It is seen that the radial stress σ_r/S_y peaks at a value of –0.6091 at θ = 0°, and monotonically reduces to zero at θ = η. At θ = 0° the tangential stress σ_t/S_y = 2.5962, and it first dips slightly before it smoothly increases to a peak value of σ_t/S_y = 2.7541 occurring at θ = η, after which it decreases to a minimum value of σ_t/S_y = –0.7947 at θ = 0°.

From these results, which are in excellent agreement with those determined by Stippes, Wilson and Krull [2], it is evident that the peak K_t of the plate with an insert fitted is 8.2% less than that for the plate with an empty hole. Hence, under an applied tensile load, the presence of the insert reduces the maximum tangential stress in the plate. Furthermore, it is expected

that cracking in the plate might tend to initiate on the hole boundary in close vicinity to the location of the peak in the tangential stress, which now occurs at $\theta = \eta$, rather than at $\theta = 0°$ as is the case for an empty hole.

It is worth noting that the gradient of σ_t/S_y is discontinuous at $\theta = \eta$, where its value abruptly changes sign from positive to negative, producing a very sharp peak. The behaviour of σ_r/S_y is such that the slope of the curve at $\theta = \eta$ is very steep, and appears to be approaching 90° to the horizontal, which produces a very rapid and step-like transition in the radial stress. If sharp and rapid changes such as these are to be accurately reproduced by an FEA model, then a highly refined mesh will need to be used in those particular regions. If this is not done, then a smoothing effect on the results can be expected, which will serve to act like a low-pass filter, removing any rapidly varying content that requires a high-density spatial mesh in order for it to be accurately resolved.

3.2 Plate and pin with different elastic material properties

In a subsequent conference paper that was published in 1964, Wilson [3] provided an analytical solution to the more general 2D contact problem of an infinite elastic plate loaded by stresses S_x and S_y at infinity and containing a smooth elastic circular insert of a different material. The solution method involved an iterative procedure and, although the solution was described as approximate, Wilson's comparisons with known analytical solutions for some special cases appear to indicate that it is nonetheless quite accurate.

Wilson [3] derived the following expressions for computing the radial and tangential stresses, σ_r and σ_t, where P_k denotes the Legendre polynomial of degree k, once the contact angle η and the superposition constants A_0, …, A_p have been determined. From his convergence studies of different test cases, it would appear that good results can be obtained with no more than five terms, corresponding to $p = 4$.

$$\begin{aligned}\sigma_r(\theta) &= 8\sqrt{\cos^2\theta - \cos^2\eta}\sum_{n=0}^{n=p} A_n \cos\left[(2n+1)\theta\right], \quad \begin{matrix} -\eta \le \theta \le +\eta \\ \pi - \eta \le \theta \le \pi + \eta \end{matrix} \\ &= 0 \text{ elsewhere}\end{aligned} \tag{4}$$

$$\sigma_t(\theta) = (S_y + S_x) + (S_y - S_x)(2\cos(2\theta) + 1) + 4B - \sigma_r(\theta)$$

where

$$B = \sum_{n=0}^{n=p} A_n D_{n+1}^*$$

$$\begin{aligned} D_0 &= 1 \\ D_1 &= -\cos(2\eta) \\ D_k &= \frac{\cos(2\eta)P_{k-1}(\cos(2\eta)) - P_k(\cos(2\eta))}{k-1}, \quad k \ge 2 \end{aligned} \tag{5}$$

$$D_{n+1}^* = \delta_{0n} + D_{n+1}, \text{ where } \delta_{00} = 1 \text{ and } \delta_{0n} = 0 \text{ for } n \ge 1$$

In order to be able to obtain a solution for the case of an aluminium plate and a titanium disk insert, a FORTRAN program was written that implemented the method described by Wilson [3]. A listing of the source code for this program can be found in Appendix B. The input parameters used by the program were the elastic material properties of the plate and insert, the stress components at infinity, and the number of terms to be used in the approximating series. The output quantities include the contact angle, the superposition constants A_0, …, A_p in the series used to approximate the contact stress, and the radial and circumferential stress around the hole boundary and also at a selected set of boundary points. Once the contact angle and constants pertaining to any combination of materials and loading are determined, the stresses themselves are reasonably amenable to being calculated in a spreadsheet, should this be desired.

The program required the use of the following numerical computations: a) Simpson's Rule for numerical quadrature; b) solution of simultaneous equations (using the method proposed by Moler [7]); c) evaluation of Legendre Polynomials of arbitrary degree k; and d) computation of the smallest root of a nonlinear equation involving said Simpson's Rule, simultaneous equations, and Legendre Polynomials in the function being solved (using a method proposed by Shampine and Watts [4]). The evaluation of one of the required integrals was complicated by the presence of singular terms, but fortunately it can be shown that these cancelled in the limit as $\theta \to 0$ (see Appendix C for details of the mathematical derivation). This enabled the requisite integral to be easily computed using numerical quadrature.

In his paper, Wilson [3] provided results for 8 test cases that involved different combinations of material properties and with the load applied exclusively in either the x-direction or the y-direction, for cases where S_x and S_y were compressive and tensile in nature, respectively. For his chosen cases, he tabulated the values of the superposition constants A_0, …, A_p, as well as giving the values of η, $\sigma_r(0°)$, $\sigma_t(0°)$, $\sigma_t(\eta)$, and $\sigma_t(90°)$. For a subset of 4 out of those 8 cases, he also provided plots of σ_r and σ_t. To verify the present program, the 8 test cases were run and the results compared to those provided by Wilson. In all cases the computed values of η and the 4 chosen stresses were in very good agreement, usually to at least three or four significant figures, sometimes more. The computed curves of σ_r and σ_t appeared to also match with the published results. However, for Case #3, the present program produced superposition constants that were almost an order of magnitude greater than those obtained by Wilson. Nevertheless, the computed values of η, σ_r and σ_t matched the published data very well. Investigation of this issue indicates that the symmetric matrix of simultaneous equations, which must be solved in order to obtain the superposition constants A_0, …, A_p, is somewhat ill-conditioned, as judged from it having quite large condition numbers, especially as p increases in size. It seems that for Case #3 this resulted in greatly different superposition constants than those published, while still obtaining what appears to be a valid solution.

For the case of an aluminium alloy plate with a titanium alloy insert, loaded uniaxially in tension ($S_x = 0$ and $S_y > 0$), plots of the normalised radial and tangential stresses, σ_r/S_y and σ_t/S_y, around the boundary of the hole are shown in Figure 4. Using $p = 4$, the contact angle was computed to be $\eta = 19.31°$, and the superposition constants were $A_0 = 0.68763$, $A_1 = -0.82157$, $A_2 = 0.59685$, $A_3 = -0.21979$, and $A_4 = 0.033940$. The peak radial stress was $\sigma_r/S_y = -0.7330$ at $\theta = 0°$, which is 20.3% greater in magnitude than if the titanium alloy insert had

instead been made from the aluminium alloy. The peak tangential stress was $\sigma_t/S_y = 2.8060$ at $\theta = \eta$, and at $\theta = 0°$ the tangential stress was $\sigma_t/S_y = 2.5105$. This indicates that the peak K_t is only 1.9% higher and the contact angle only 0.31° less than if the titanium alloy insert had instead been made from the aluminium alloy, even though the stiffness of the titanium alloy is 65% greater than that of the aluminium. Compared to the results shown in Figure 3, as a result of its much greater stiffness than the aluminium plate material, the titanium insert also produces a somewhat greater reduction in σ_t along the contact arc relative to the peak stress in the distribution than did the aluminium insert, by approximately 5% or so.

4. General finite element analysis approach

The Abaqus 6.9-1 FEA code was used to perform the 2D and 3D analyses that are reported here, and Abaqus/CAE 6.9-1 was used as the pre- and post-processor. For the purposes of creating the geometry in Abaqus, an orthogonal right-handed x-y-z coordinate system was defined with its origin at the geometric centre of the coupon, as shown in Figure 1. The horizontal x-axis was aligned parallel to the transverse direction of the coupon, the vertical y-axis was aligned parallel to the longitudinal direction of the coupon, and the z-axis was aligned parallel to the thickness direction of the coupon. This is represented by the idealised general 3D plate of height H and width W as shown in Figure 5, which has a uniaxial load S applied in the y-direction. To help reduce the size of each finite element model, and hence reduce the computation times, ¼-symmetry was utilised when creating a finite element mesh for the 2D problems, and ⅛-symmetry was used for the 3D problems.

4.1 Considerations when analysing infinite-plate benchmark problems

In this report, FEA techniques are being utilised with a view to performing contact analysis of 3D physical structures that have finite dimensions. As part of the verification process used to establish that the FEA solution techniques being used are giving good results, whenever possible it is highly desirable to be able to compare FEA results to known analytical solutions. A good match here will provide confidence that the FEA formulation is accurate and is also being correctly used.

For the contact problem at hand, Section 3 described some 2D analytical solutions for infinite plates with inserts that were similar to that occurring for the coupon with a neat-fit fastener inserted in the hole in the coupon. 2D FEA techniques can be used to model these problems, with the limitation that a finite-width plate needs to be used of necessity. When computing SCFs for holes in plates, the finite-width nature of practical problems is well known, and indeed results are available that embody suitable correction factors [8, 9, 10]. However, for the contact problem presently under consideration here, no such finite-width corrections exist.

In the present benchmarking work, the finite-width affected FEA solutions will need to be compared directly to the available infinite-plate contact solutions. In that case, it is necessary to choose the relative plate–hole dimensions so as to minimise the effects of finite plate width on the solution. Hence, a number of FEA simulations were conducted on a square plate, of width W, with a central circular hole of diameter D, loaded by a far-field uniaxial stress equal to S_y, in order to investigate the convergence of the $K_t = \sigma_t(0°)/S_y$ value to the theoretical 2D

infinite-plate ($W/D = \infty$) solution value of $K_t = 3$ [6], where the equation for the tangential stress around the circumference of the hole is given by:

$$\sigma_t(\theta) = S_y(1 + 2\cos(2\theta)) \tag{6}$$

For maximum accuracy, 8-noded quadratic elements were used in the FEA. Two different levels of mesh refinement were studied, one being a coarse-mesh model that used only 20 equispaced elements around the ¼-circumference of the hole (see Figure 6), and the second being a fine-mesh model with 180 equispaced elements (see Figure 7). A range of plate–hole configurations with different W/D ratios were studied, and the results corresponding to these two levels of mesh refinement and $W/D = 40$ are presented in Table 1. The 180-element solution gave $K_t = 3.00539$, which is within 0.18% of the exact infinite-plate analytical solution. Even the 20-element mesh produces a K_t that is within 0.47% of the infinite-plate analytical solution. A comparison of the distribution of tangential stress calculated from the analytical solution as compared to the FEA results obtained for the 20-element and 180-element cases is shown in Figure 8, and it is evident that excellent agreement has been attained with both levels of mesh refinement.

When proceeding to analyse benchmark infinite-plate contact problems using FEA, it is therefore concluded that using a plate–hole configuration with $W/D = 40$ should serve to enable direct comparison of the FEA results with the analytical solution. This is because the finite-width effects in a quasi-infinite plate such as this with $W/D = 40$ are reduced to a very low level, well under 0.5% for even a relatively coarse mesh. Of course, for the contact problems of interest, as a result of the discontinuous behaviour of the radial and tangential stresses in the vicinity of the end of the contact arc at $\theta = \eta$, the highly-refined 180-element mesh will better serve to reproduce with high fidelity the observed known behaviour.

4.2 Contact modelling assumptions

This is a mixed boundary condition problem with moving boundaries, where the surfaces of the insert and the hole in the plate can come into contact with each other. The broad assumptions that are used are:

- Linear-elastic, isotropic, homogenous materials.
- Zero friction.
- Small sliding.
- In-plane remote uniaxial tension loading applied to ends of the plate (35 kN load).
- No pin loading.
- No compression loading.
- Both the pin and the hole can deform during contact.
- Non-advancing contact behaviour (contact area does not vary with load).
- Augmented Lagrange contact constraint enforcement method.
- “Hard” contact pressure–overclosure relationship.

The results of this work are valid up to the elastic limit of the material. Any results beyond the elastic limit of either the coupon or pin material, though not entirely accurate, may be useful as an estimate for the general behaviour of the coupon–pin combination at higher loads.

5. 2D FEA of analytical filled-hole contact problem

In order to verify the ability of Abaqus to obtain accurate solutions to 2D contact problems, it was chosen to model a square aluminium plate of width W with a neat-fit titanium pin inserted into the hole of diameter D. The ratio of plate width to hole diameter, W/D, was chosen to be relatively large at $W/D = 40$ in an attempt to create a quasi-infinite plate to reduce finite-width effects to negligible levels. The 2D FEA model utilised ¼-symmetry, as well as two levels of mesh refinement to see what effect this might have on predictions of the stress distribution around the boundary of the hole, both on and beyond the contact arc. The frictionless surface-to-surface contact model available in Abaqus was used for analysing the hole–pin contact behaviour, and the Abaqus default analysis parameter settings were used.

The coarse-mesh model that was created had 20 equispaced elements around the ¼-circumference of the hole–pin boundary, while the fine-mesh model had 180 equispaced elements. Details of the two meshes in the vicinity of the hole–pin boundary are shown in Figure 9. The hole–pin interface is indicated there, with the finite elements for the pin being shown as shaded, and the origin of the global x-y coordinate system is located at the centre of the pin as shown. As 8-noded quadratic elements were used, the fine-mesh model has a nodal spacing interval of 0.25° around the hole boundary, which will assist in resolving fine details. On the other hand, the coarse-mesh model has a nodal spacing of 2.25°, which is expected to produce considerable smoothing around any rapid changes in the radial and tangential stresses.

The 2D FEA results for the normalised radial and tangential stresses for the filled-hole case, σ_r/S and σ_t/S, where $S = S_y$, are shown in Figure 10. The results obtained using the 2D analytical solution proposed by Wilson [3] are also shown, and the empty-hole analytical solution is also provided for reference. It is clear that the fine-mesh 2D quasi-infinite plate results are in excellent agreement with the 2D infinite-plate analytical contact solution. The peak $\sigma_t/S = 2.7957$ is located at $\theta = 19.75°$ and is very well resolved with only a small degree of rounding evident. It is only 0.4% less than the analytical result of $\sigma_t/S = 2.8060$ located at $\theta = 19.31°$. The coarse-mesh results for σ_t/S are also quite good, but the sharp peak in σ_t/S has been smoothed over and shifted in location to $\theta = 20.25°$, the value of $\sigma_t/S = 2.7332$ being smaller in magnitude by about 2.6% compared to the analytical result. The coarse-mesh results for σ_r/S agree moderately well with the analytical solution, but the differences in the vicinity of $\theta = \eta$ are quite noticeable. Once the length of the contact arc is identified with a reasonable degree of precision, it would of course be possible to further refine the mesh around that location, and this would be expected to enhance the accuracy of the FEA results.

6. 2D FEA of LIF Hawk Filled Hole Coupon contact problem

Considering the experience gained from analysing the contact problem involving the quasi-infinite plate, two 2D FEA models of the LIF Hawk Filled Hole Coupon were created using two levels of mesh refinement around the hole–pin contact boundary. As before, ¼-symmetry and 8-noded quadratic elements were used. The coarse-mesh and fine-mesh models utilised 20 and 180 equispaced elements distributed around the ¼-circumference of

the hole–pin interface. The coarse-mesh model is shown in Figure 11a, where the full model is presented on the left and a detail view of the mesh around the contact boundary is on the right. Similarly, the fine-mesh model is shown in Figure 11b. The shaded finite elements correspond to those being used to model the titanium pin. Broadly speaking, these two meshes are equivalent to the ones that were used when analysing the quasi-infinite plate in the previous section. The analysis of the fine-mesh model was completed in about 15 seconds.

Plots of the variation in the normalised radial and tangential stresses, σ_r/S and σ_t/S, are shown in Figure 12 for both the fine-mesh and coarse-mesh models, as well as the finite-width-corrected results for the analytical infinite-plate solution. The finite-width correction (FWC) factor was arrived at by taking the ratio of the peak σ_r/S stress obtained from the 2D FEA of the coupon with an empty hole (no pin fitted) and the infinite-plate analytical solution. This gave a FWC factor of 3.1786/3 = 1.0595, which was then used to scale up the infinite-plate analytical results for this hole–pin configuration. For reference, Figure 12 also shows the variation of σ_t/S for the empty-hole infinite-plate analytical solution as well as for the FEA of the empty-hole coupon. Selected values of σ_r/S and σ_t/S at different values of θ around the hole boundary, including $\theta = \eta$, are presented in Table 2.

It is worth comparing the K_t of the empty-hole coupon, K_{teh} = 3.1786, with that obtained from handbook values. From the data presented in Chart 4.1 in *Peterson's Stress Concentration Factors* [9] for a finite-width thin plate with a circular hole, the expression for K_{tg} in terms of plate width W and hole diameter D is:

$$K_{tg} = \frac{2 + 0.284\left(1-\frac{D}{W}\right) - 0.600\left(1-\frac{D}{W}\right)^2 + 1.32\left(1-\frac{D}{W}\right)^3}{\left(1-\frac{D}{W}\right)} = \frac{3.004 - 2.476\left(\frac{D}{W}\right) + 3.36\left(\frac{D}{W}\right)^2 + 1.32\left(\frac{D}{W}\right)^3}{\left(1-\frac{D}{W}\right)} \tag{7}$$

Note that the above formula has a very small but finite error when $D/W = 0$, giving K_{tg} = 3.004 instead of 3 for this limiting case.

Pilkey [10] provides a similar formula in Table 6.1, with his formula giving the correct value of $K_{tg} = 3$ when $D/W = 0$. Pilkey's formula is:

$$K_{tg} = \frac{3.000 - 3.140\left(\frac{D}{W}\right) + 3.667\left(\frac{D}{W}\right)^2 - 1.527\left(\frac{D}{W}\right)^3}{\left(1-\frac{D}{W}\right)}, \quad 0 \le \frac{D}{W} \le 1 \tag{8}$$

For the coupon geometry at the minimum cross-section, W = 30 mm and D = 6.35 mm (D/W = 0.2117), and Peterson's formula gives K_{tg} = 3.168, while Pilkey's formula gives K_{tg} = 3.153. Both of these results are slightly less than the 2D-FEA-computed value of 3.1786 for the empty-hole coupon, which is likely due to the fact that we are dealing with a dog bone–shaped coupon rather than a simple straight-sided strip. The above formulas produce values of K_{tg} that are similar to those that were provided by Howland [11] for discrete values of D/W = 0, 0.1, 0.2, 0.3, 0.4, and 0.5.

From an inspection of Figure 12 and Table 2, it is clear that results produced by the fine-mesh and coarse-mesh FEA models of the coupon with pin fitted are in quite good agreement with each other. As anticipated, the coarse mesh once again produces a peak in σ_t/S at $\theta = \eta$ that is smoothed out and lower in magnitude, as well as having a slightly higher value of η. It is also evident that the analytical infinite-plate solution with a FWC factor applied is also a good match for the fine-mesh coupon results.

A number of stress contour plots have also been generated for an applied load of 35 kN, which corresponds to an applied stress of 194.44 MPa. Figure 13a shows the results for radial stress σ_r, Figure 13b shows the results for the tangential stress σ_t, and Figure 14 shows the results for the Von Mises stress. As expected, the plots of the tangential and the Von Mises stresses clearly indicate the location of the hole–pin interface, as a result of the Young's modulus of the pin and the plate being so different. Also as expected, the plot of the radial stress shows continuity of stress across the hole–pin interface. Looking at Figure 14, it appears to be likely that a significant degree of plasticity could be expected in the aluminium plate along the entire contact arc, as at this 35 kN load level there is a large region of material where the Von Mises stress has exceeded the yield strength of the aluminium alloy material (which is approximately 400 MPa). However, an elasto–plastic analysis of this behaviour will not be performed here, but will be left for a subsequent separate investigation.

7. 3D FEA of LIF Hawk Filled Hole Coupon contact problem

As was done when analysing the 2D contact problem using FEA, it was decided to model the 3D coupon using a coarse mesh as well as a fine mesh, with a view to obtaining some insights into the quality of results that could be achieved. However, because the computational complexity of 3D FEA models is much greater than that of 2D models, owing to the greatly increased number of degrees of freedom that need to be solved for, the level of mesh refinement that can usefully be utilised is considerably less than what is typically possible with 2D models. Nonetheless, it was decided to attempt to use a very refined mesh in order to try to very accurately resolve the sharp transitions in tangential and radial stress that were evident in the 2D analysis, and which were therefore expected to also appear in the 3D analysis.

As in the 2D analyses, the frictionless surface-to-surface contact model available in Abaqus was used for analysing the hole–pin contact behaviour, and the Abaqus default parameter settings were used. The Abaqus documentation recommends that the master contact surface consist of the more rigid and/or more highly refined surface. Hence, as the mesh densities

used on the pin and the hole were quite similar, the master surface was defined to be the pin, which is made from titanium and is about 65% stiffer than the aluminium material from which the coupon is manufactured. The slave contact surface was defined to be the surface of the hole.

7.1 3D coarse-mesh finite element model

7.1.1 Meshing considerations

The coarse-mesh FEA model that was created for analysing the aluminium coupon and titanium pin (bolt) combination is shown in Figure 15. The upper picture shows the general mesh of the entire ⅛-symmetry model, and the lower picture shows a detail of the mesh in the immediate vicinity of the hole. The pin is nominally modelled as a hand-tightened loose-fit bolt in a hole, without any of the restraints that normally would apply to a bolt head and nut combination in a fully torqued-up bolt. In the FEA model, the pin extends 1 mm beyond each of the outer longitudinal surfaces of the plate. To some degree, this is anticipated to simulate the presence of the additional material that is associated with the bolt at the head and nut ends.

The coarse-mesh model was predominantly composed of 20-noded quadratic C3D20 hexahedral brick elements. Some 15-noded quadratic C3D15 elements were also present. In developing the mesh, the coupon was partitioned into subregions, and use was made of structured as well automated swept meshing. There were 12 equispaced elements distributed over the half-thickness of the plate (4 elements per mm). The hole had 16 elements distributed around the ¼-circumference of its boundary, and a 3:1 mesh bias was used to provide some mesh refinement in the region where the peak stress occurs. A total of 6592 elements were defined using 23266 nodes. There were 61386 variables present in the model. The 35 kN load was applied as a uniform pressure of 194.44 MPa over the faces of the solid elements located along the bottom of the coupon mesh.

7.1.2 Computed stress distributions

The stress contour plots for the tangential and radial stresses in the vicinity of the hole in the coupon are shown in Figure 16, where the stresses are presented in MPa. The tangential stresses are highest at the midplane of the coupon, and reduce progressively towards the free surface of the coupon. This is not unexpected, as it is relatively well known [12, 13, 14, 15, 16] that the through-thickness tangential stress monotonically decays along the bore of an empty hole in a uniaxially-loaded plate for $D/t > 0.5$ (for our coupon $D/t = 1.0583$). In contrast to this behaviour, at the location where the free surface of the coupon meets the pin, there was a high stress concentration evident in the radial stresses (see Figure 16b). Upon closer examination, this appeared to indicate the presence of a stress singularity. Phenomena such as this, where a singular stress field is developed at the vertex of an elastic plane indenter of various angles that is compressing another elastic plane, have been the focus of extensive work by many authors. That work has included analytical studies [17, 18, 19, 20], experimental studies using photoelasticity [21, 22, 23, 24], and FEA studies [24, 25, 26, 27]. The singularity appears to have a strength that is approximately of the order of $1/\sqrt{r}$.

7.1.3 Additional meshing considerations due to presence of stress singularity

Having identified that the 3D elastic coupon–pin contact problem involves a singularity, it is evident that the coarse-mesh model will not be able to provide sufficiently accurate estimates of the stress in the singularity-affected region. Sinclair *et al*. [26] provide some guidance as to why the singular stress field should be accurately represented, using the following example:

> "*… we observe that it should not be thought that the smoothing of stress gradients which accompanies plastic flow obviates the stress analyst from accurately resolving elastic stress fields if accurate elasto–plastic stresses are sought. Basically this is because elastic response physically precedes and triggers elasto–plastic. To explain further, using the fine submodel grid, local first yielding for the dovetail without friction is predicted to occur when loads attain 58 percent of the maximum value used here (based on a Tresca yield criterion). Using just the coarse global grid, this event is not predicted to occur until loads reach 84 percent of their maximum value. Clearly a significant erroneous delay results from using a finite element mesh of insufficient refinement.*"

By choosing a suitable level of mesh refinement, it is anticipated that the accuracy of the present elastic analysis will be enhanced. It is also considered that any subsequent elasto–plastic analysis of the coupon–pin combination will more accurately simulate the development of the plastic zone with increasing load. In an empirical study, Whitcomb, Raju, and Goree [28] concluded that finite element solutions are accurate everywhere except very near a stress discontinuity or singularity, and that the region of inaccuracy is limited to about two elements in the immediate vicinity thereof. This region of inaccuracy can then be made very small by progressive mesh refinement, such that valid results can be obtained by FEA in the neighbourhood of stress discontinuities and singularities. As will be demonstrated later, the results of the present work appear to support this two-element accuracy rule of thumb.

7.2 3D fine-mesh finite element model

7.2.1 Meshing considerations

A fine-mesh FEA model was created for analysing the aluminium coupon and titanium pin combination to provide a point of reference against which less refined meshes could compared. The fine-mesh model is shown in Figure 17, and it uses significantly more elements than does the coarse-mesh model. The upper picture shows the general mesh of the entire ⅛-symmetry model, and the lower picture shows a detail of the mesh in the immediate vicinity of the hole. There were 60 equispaced elements distributed over the half-thickness of the plate (15 elements per mm), and the hole had 90 equispaced elements distributed around the ¼-circumference of its boundary (18 elements per mm). Hence, the elements around the hole boundary were well-shaped, being of approximately of 1:1 aspect ratio, which helps to enhance the accuracy of the simulations.

7.2.2 Computed stress distributions

The stress contour plots for the tangential and radial stresses in the vicinity of the hole in the coupon are shown in Figure 18, where the stresses are presented in MPa. Although the fine-mesh model appears to be quite capable of providing high-fidelity results, this model takes a

very long time to run, approximately 18 hours. It is therefore deemed to be impractical to use for any nonlinear plasticity analysis, which requires the use of multiple load increments and iterative plasticity solutions at any given load increment, on top of any contact iterations. On the other hand, the coarse-mesh model has been judged to be somewhat too coarse, especially in view of the fact that the radial stress in the hole exhibits a singularity where the outer surface of the plate meets the surface of the pin, at $z/t = \pm 0.5$. As a result of these considerations, it was decided to investigate the use of a graded mesh to maintain accuracy in the solution while reducing the run time.

7.3 3D graded-mesh finite element model

7.3.1 Meshing considerations

A graded-mesh FEA model was created for analysing the aluminium coupon and titanium pin combination, where the degree of mesh refinement was somewhere between that of the fine-mesh and coarse-mesh models described earlier. The graded-mesh model is shown in Figure 19. The upper picture shows the general mesh of the entire ⅛-symmetry model, and the lower picture shows a detail of the mesh in the immediate vicinity of the hole. There were 14 elements distributed over the half-thickness of the plate, and the elements were made smaller towards the free surface of the plate, with the last two elements being equispaced. The hole had 20 elements distributed around the ¼-circumference of its boundary, and the mesh was graded so that it was finer at the two ends of the arc around the ¼-circumference (at $\theta = 0°$, 90°, 180° and 270° around the complete hole), and coarsest in the middle of that arc (at $\theta = 45°$, 135°, 225°, and 315° around the complete hole). This refinement occurs in the regions where stresses will be high as a result of contact occurring under tension loading (the present case) and compression loading.

7.3.2 Computed stress distributions

The stress contour plots for the tangential and radial stresses in the vicinity of the hole in the coupon are shown in Figure 20, where the stresses are presented in MPa. The contour plot of Von Mises stress is shown in Figure 21. This model took approximately 15 minutes to run, and it appears to give similar results to the fine-mesh model. The singularity in the radial stress appears to be represented more accurately than was possible with the coarse-mesh model. It is worth noting that, although the radial stress at the free boundary in the contact region will be high, being radial and compressive in nature rather than tangential and tensile in nature, means that it will be of low significance from a fatigue cracking point of view. Furthermore, the effects of plastic flow at higher load levels can be expected to ameliorate the effects of the stress singularity; it is planned to study this at a later date in another report.

8. Discussion of 2D and 3D FEA results

8.1 Identification of region of substantial plasticity

All of the different linear-elastic FEA results were obtained for a load level corresponding to an applied uniaxial load on the coupon of 35 kN. In order to put the magnitude of this applied load in perspective, consider that, at a 25 kN load level, local yielding of the open

hole is just starting to occur between 405–450 MPa. The average gross-section stress (hole excluded) is about 139 MPa and the average net-section stress (hole included) is about 176 MPa. Loads above about 25 kN will produce strains beyond the material elastic limit for the aluminium alloy coupon. The contour plot of Von Mises stress shown obtained from a 3D FEA is shown in Figure 21, and corresponds to the 35 kN linear-elastic load level. This indicates the existence of a plastic zone that will be developed over a large portion of the hole surface, and the size of this plastic zone includes all of the hole–pin contact interface (Von Mises stress contours above 405 MPa).

8.2 Comparison of 2D and 3D FEA results with 2D analytical solution

The variation in the radial and tangential stresses around the hole boundary, as obtained from the 3D and 2D Abaqus FEA fine-mesh midplane results and the 2D analytical infinite-plate finite-width-corrected solution, is shown in Figure 22. For the radial stress, σ_r/S, determined at the midplane (z/t = 0.0), the 3D FEA results in Figure 22a match the 2D benchmarks very well at the point of peak stress at θ = 0°, but the agreement gets progressively worse as $\theta \rightarrow \eta$, although the general shapes of the curves are quite similar. The 3D FEA result for the angular length of the contact arc is η = 19.00°, which is about 0.75° less than the result obtained from the 2D FEA of the coupon. Turning now to the tangential stress, σ_t/S, determined at the midplane, the 3D FEA results in Figure 22b are again in relatively good agreement with the 2D benchmarks, with the general shapes of the curves matching up quite well. The 3D FEA contact solution shows the distinct flat shelf in the σ_t/S stress distribution in the range $0° \le \theta \le 0.7\eta$, followed by the characteristic rise and fall as θ approaches and then exceeds η in value. Comparing the results from the open-hole case with those from the contact case, the values of σ_t/S are less than those for the open-hole case until about θ = 18° ($\approx 0.95\eta$). Beyond this point the tangential stress response curve is shifted to the right, resulting in the positive values of tangential stress being higher in value, and negative values being lower. The effect of contact has been to reduce the peak value of σ_t/S from 3.347 to 3.031, which is a significant reduction of 9.4%.

8.3 Accuracy of graded-mesh finite element model

The variation of the radial stress and tangential stress around the hole at both the midplane (z/t = 0.0) and the surface (z/t = 0.5) in the uniaxially-loaded aluminium coupon with a neat-fit titanium pin inserted, as obtained from the 3D FEA, is shown in Figure 23a and Figure 23b, respectively. These results compare the solutions obtained using the coarse-mesh, fine-mesh and graded-mesh FEA models, where the graded mesh is itself only marginally more refined than is the coarse mesh. Using the fine-mesh model as the reference yardstick, it is evident that the graded-mesh model produces a much better match in the radial and tangential stresses than does the coarse-mesh model. This is particularly evident for the radial stresses that occur at the vertex formed by the free surface of the coupon where it meets the surface of the pin. As a result of the stress singularity that occurs there, which has been previously discussed, the values of σ_t/S for the coarse-mesh model are significantly lower than the predictions from the fine-mesh model, while those for the graded-mesh model are much closer as a result of using a finer mesh near the free surface of the coupon.

Looking at Figure 23b, it is quite apparent that neither the graded-mesh nor the coarse-mesh models can reproduce the sharp peak in the σ_t/S distribution that normally occurs at $\theta = \eta$. This occurs because their meshes are simply too coarse to resolve the fine details in a region where there is a very rapid change in sign in the slope of the curve at $\theta = \eta$, resulting instead in a very smoothed approximation of what is in actuality a quite sharp and pointy peak. Even the fine-mesh model struggles somewhat in this regard, producing some rounding of the peak. When compared to the coarse-mesh model, the graded-mesh model exhibits an interesting step-like change in σ_t/S at the free surface in the region $0° \le \theta \le 0.7\eta$, producing results that are in good agreement with those from the fine-mesh model. Although this step-like reduction is only of the order of 3% or so, it is nonetheless a very noticeable feature. Even though the radial contact pressure σ_r/S is orthogonal to the tangential stress σ_t/S, the singular behaviour of σ_r/S nonetheless appears to have a noticeable effect on the tangential stress occurring on the free surface if a coarse mesh is utilised.

8.4 Through-the-thickness stress distribution effects

The variation of the radial stress and tangential stress along the bore of the hole in the aluminium coupon fitted with a neat-fit titanium pin is shown in Figure 24. These results were obtained from the 3D FEA along a line of constant $\theta = 0°$ over the interval $0 \le z/t \le 0.5$. It is apparent that the radial stress increases from the midplane ($z/t = 0$) towards the outer free surface of the coupon ($z/t = 0.5$), while the tangential stresses do the opposite, decreasing in value from the midplane to the free surface. As mentioned previously, the peak tangential stress is known to vary along the bore of an empty hole in a thick uniaxially-loaded plate [12, 13], and the variation is a function of the ratio of the hole diameter to the thickness of the plate, D/t, as well as Poisson's ratio, υ. The results presented by Folias and Wang [12] indicate that, for ratios where $D/t > 0.5$, the peak tangential stress along the bore of an empty hole will decrease monotonically from the midplane to the outer free surface of the plate. For the present coupon, we have that $D/t = 1.0583$, and the tangential stress along the bore depicted in Figure 24b reduces in the manner commensurate with their predictions. It is expected that a similar trend would be present had a curved line down the bore corresponding to $\theta = \eta$ been used (i.e. corresponding to the peak in the tangential stress distribution), and this will be confirmed below.

As shown in Figure 24, both the radial and tangential stress distributions exhibit a distinct step-like change in response between the results from the coarse-mesh and graded-mesh models. It seems reasonable to attribute this step change to the local refinement of the graded-mesh model in the vicinity of the stress singularity in σ_r/S that occurs at $z/t = 0.5$. It is interesting to see that the step change is active over the entire region, whereas the extra mesh refinement in the graded-mesh model is only over the last few percent of the thickness of the coupon. The stress predictions obtained from the graded-mesh model are in very good agreement with those from the fine-mesh model up to the final two elements terminating at $z/t = 0.5$, which supports the use of the two-element rule of thumb described earlier.

Looking at Figure 24b, it is also interesting to note that the tangential stress σ_t/S takes a rather sudden dip in the vicinity of the singularity in σ_r/S, dropping from a value of 2.529 at $z/t = 0.4875$ to 2.317 at $z/t = 0.5$, a reduction of about 8.3% in the space of $\Delta t/t = 0.025$ (2.5% of the plate thickness). No such sudden change was indicated by the work of Folias and

Wang [12] in determining the 3D stress field around a circular hole in an infinite plate of arbitrary thickness. Their results for a plate with $\upsilon = 0.33$ and $D/t = 1.0$ are plotted in Figure 25. Also shown there are the 3D FEA results for the coarse-mesh model of the coupon with an empty hole (no pin inserted), together with the analytical results modified by the application of the FWC factor of 1.0597 obtained from the 2D FEA of the empty-hole coupon. Here the peak tangential stress from the 3D FEA of the open-hole coupon is $\sigma_t/S = 3.347$, and applying the inverse of the 2D FEA-computed FWC we obtain $\sigma_t/S = 3.158$, which is 0.7% less than the value $\sigma_t/S = 3.18$ obtained by Folias and Wang [12].

Considering that $D/t = 1.0583$ for the present coupon, while the analytical results pertain to $D/t = 1$, the 3D FEA and FWC-factored 2D analytical results are in quite good agreement. Better agreement would potentially have been obtained had the analytical solution corresponding to $D/t = 1.0583$ been available, as the data presented in Figure 6 of Folias and Wang [12] indicates that the peak K_t at $z/t = 0$ would be reduced a little, while the minimum K_t at $z/t = 0.5$ would be increased a little, for that particular case.

The 3D FEA results obtained for the radial and tangential stresses at selected angles around the hole–pin contact interface at the midplane and surface faces of the coupon for different levels of mesh refinement are presented in Table 3. It is also noted here that η varies along the bore of the hole. For the fine-mesh FEA results, the value of η is smallest at the midplane ($\eta = 19.0°$), swinging around by about 2.5° (to $\eta = 22.5°$) as the free surface of the coupon is approached. This general behaviour is also evident in the coarse-mesh and graded-mesh results.

8.5 Comparison of midplane and free surface stress distributions

Looking at the midplane and free surface tangential stress distributions computed using 3D FEA and shown in Figure 23 and Figure 24, it is evident that the midplane results are the ones that are in best agreement with the 2D analytical solution. This interesting behaviour warrants further elaboration here. For 2D plane elasticity problems, the in-plane stresses associated with states of plane stress and plane strain are in fact identical. This feature was utilised by Sternberg and Sadowsky [13], who developed an approximate 3D solution for the stress distribution around a circular cylindrical hole in an infinite plate of arbitrary thickness. They did this by obtaining correction terms to the underlying 2D equations of plane stress that are independent of plate thickness, noting that their correction terms may be regarded as 3D corrections characterising the departure from plane stress owing to $t/\mathrm{D} \neq 0$. In the limit as $t/\mathrm{D} \rightarrow 0$, their solution correctly reduces to the plane-stress solution. We note here that, using the data for $\theta = 0°$ and $\theta = \eta$ from Table 3, the 3D FEA tangential stress results at the free surface face are nominally about 15% less than those at the midplane face.

8.6 Accuracy of results surrounding the sharp peak in tangential stress

A major feature of the contact problem being considered here is that a very sharp peak in the tangential stress develops at the end of the contact arc. It is possible to obtain accurate stress results in this region, but only if a sufficiently high level of mesh refinement is utilised. The best results in this regard were obtained using an element spacing of one element for every 0.25° of arc when solving 2D problems. However, this spacing is too fine for use on typical

3D problems, as it results in models with very large numbers of elements that take tens of hours to solve for just one linear-elastic load case. A graded mesh, comprised of 20 elements distributed around the ¼-circumference of the hole, and refined near the tension and compression stress peaks, can be used with good results. It offers more modest solution times that would enable elasto–plastic analyses to be conducted in a reasonable time. However, as a result of the coarseness of the mesh relative to the significant stress features, some smoothing of the main peak in the tangential stress will occur, with a small reduction in the predicted value of the peak stress, as well as a small shift in its location.

9. Summary of key results

The key achievements of the present work are summarised below:

a. The 2D analytical solution by Stippes, Wilson and Krull [2] for computing the radial and tangential stresses around a hole in a plate containing a circular insert with the same elastic material properties has been identified. The equations for the tangential and radial boundary stresses have been coded up into VBA Excel spreadsheet functions (see Appendix A), which are now available for future use in providing initial estimates of contact stresses for applicable two-dimensional contact problems.

b. The approximate 2D analytical/numerical solution by Wilson [3] for determining the radial and tangential stresses around a hole in a plate containing a circular insert with different elastic material properties has been identified. A FORTRAN program for computing the equations for the boundary stresses using general user-supplied elastic material properties (Young's modulus and Poisson's ratio) has been written (see Appendix C). It is available for future use in providing initial estimates of contact stresses for any applicable two-dimensional contact problems.

c. The use of the Abaqus FEA code for solving 2D hole–insert contact problems using quadratic 8-noded elements has been verified by benchmarking against a known 2D analytical/numerical solution by Wilson [3]. The results indicate that FEA can be used to accurately solve problems involving contact between the boundary of a hole and an insert in a plate loaded uniaxially in tension.

d. The use of the Abaqus FEA code for solving 3D hole–insert contact problems has also been verified by benchmarking the FEA solution against a known approximate 2D analytical/numerical solution by Wilson [3]. The geometry and aluminium and titanium elastic material properties representative of a fatigue test coupon were used. Both coarse-mesh and fine-mesh solutions were investigated, with the latter producing better, more accurate results, just as expected. As expected, the 3D midplane results from the FEA provided the best match with the 2D solution. This is because the state of stress at the central plane of the present finite-thickness plate (D/t = 1.0583) approaches one that corresponds to plane strain conditions [13].

e. A computationally-efficient and accurate graded-mesh 3D FEA model of the fatigue test coupon and fastener combination has been created. As reported here, tension-only loadings have been studied, but the graded-mesh model was designed to also

be suitable for the analysis of compressive loadings, which will be of interest in any potential follow-on elasto–plastic contact analysis studies that may be undertaken.

f. A mesh refinement strategy to ameliorate the effects of the radial stress singularity on the results has been identified for use in 3D FEA of hole–insert contact problems. The singularity occurs where the surface of the insert meets the free outer surfaces of the plate. The strategy makes use of the "2-element rule", where the FEA stress predictions are accurate up to the last two elements that are placed in the immediate vicinity of the stress singularity [28], and it requires that a high level of local mesh refinement be used at the stress singularity.

g. The radial and tangential contact stresses both include stress discontinuities at the end of the contact arc (at $\theta = \eta$). The accuracy with which the stresses in these transition regions can be represented is highly dependent on the level of mesh refinement that is used. When using quadratic elements, a node-to-node spacing of 0.25° (720 elements) around the full boundary of the hole has been found to give very good results. Such a choice allows the value of η to be accurately determined to the nearest 0.25° or so, as well as allowing the sharp peak in the tangential stress distribution to be accurately reproduced, together with the sudden drop to zero in the radial stress distribution.

h. The 3D coarse-mesh and graded-mesh models both produce reasonably accurate stress results at the midplane of the plate, even though the node-to-node spacing when quadratic elements are used is about 2.25° (80 elements) around the full boundary of the hole. However, there is considerable smoothing of the peak in the tangential stress distribution, which is usually accompanied by a reduction in the peak value by 2–3%, as well as a small shift in its predicted location of about 1° or so.

i. For a neat-fit titanium insert and an aluminium coupon combination undergoing predominantly tensile loading, it is envisaged that cracking in the coupon is likely to occur on the hole boundary in close vicinity to the location of the peak in the tangential stress, which occurs at $\theta = \eta \approx 19°$ (¼-symmetry assumed), rather than at $\theta = 0°$ as is the case for an empty hole. However, the location of any cracking will still be greatly influenced by where the worst initial discontinuity or manufacturing flaw is present.

j. During the fatigue testing program, the maximum load applied to the coupon is 35 kN. The linear-elastic FEA work performed here indicates that significant plastic deformation can be expected around the hole boundary, as the computed Von Mises stress levels exceed the yield point of the aluminium alloy from which the coupon is manufactured. Hence, an elasto–plastic contact analysis needs to be undertaken in order to ascertain the effects of plasticity on the stress concentration behaviour of the hole when it is fitted with the titanium pin.

10. Conclusion

A series of 2D and 3D contact analysis models of a plate with a circular hole fitted with a neat-fit circular insert, where the plate is loaded in uniaxial tension only, have been analysed in order to check the accuracy of FEA results versus a solution computed using a known analytical/numerical technique provided by Wilson [3]. The radial and tangential stresses around the hole boundary resulting from contact between the surfaces of the hole and the insert were determined, and the FEA models utilised various degrees of mesh refinement to help ascertain the effect on the accuracy of the FEA predictions.

The results show that FEA is well suited to the solution of both 2D and 3D contact problems involving materials with different elastic material properties. It is capable of producing accurate stress distributions for these types of problems, as long as a suitable level of mesh refinement is used in order to capture some of the rapid variations in stress that occur as a result of hole–insert contact interactions.

The results from the analysis of the fatigue test coupon reported here provide a body of improved stress distribution data for use in the validation of test interpretation activities relating to full-scale fatigue testing of aircraft structures in service with the RAAF. As it has been determined that the peak stresses around the hole considerably exceed the yield point of the aluminium alloy coupon material, it is recommended that an elasto–plastic finite element contact analysis be undertaken. This would help to quantify the effects of local plasticity on the stress concentration factor, as the effects of plastic flow can be anticipated to have a significant effect on fatigue life.

11. Acknowledgement

The author would like to thank Dr Colin Pickthall, a work colleague, for his valuable insight and assistance in providing mathematical advice relating to handling the presence of singularities in one of the integrals that needed to be computed.

12. References

1. Brombolich, L. J. Elastic-plastic analysis of stresses near fastener holes. AIAA Paper No. 73-252, AIAA 11th Aerospace Sciences Meeting, Washington, D.C., 10–12 January 1973.

2. Stippes, M., Wilson, H. B., Jr., Krull, F. N. A contact problem for a smooth disk in an infinite plate. *Proceedings of the Fourth US National Congress of Applied Mechanics*, Vol. 2, 1962, pp. 799–806.

3. Wilson, H. B., Jr. Approximate determination of contact stresses in an infinite plate with a smooth circular insert. *Proceedings of the 2nd Southeastern Conference on Theoretical and Applied Mechanics*, 5–6 March 1964, Atlanta, Georgia, USA.

4. Shampine, L. F., Watts, H. A. FZERO, a root-solving code. Report SC-TM-70-631, Sandia National Laboratories, September 1970.

5. Thatcher, H. C. Algorithm 165, Complete Elliptic Integrals. *Communications of the ACM*, Volume 6, No. 4, April 1963, pp. 163–164.

6. Timoshenko, S. Theory of Elasticity. The effect of circular holes on stress distribution in plates, Section 35, pp. 90–97, McGraw-Hill, Singapore, 1982.

7. Moler, C. B. Algorithm 423: Linear Equation Solver. *Communications of the ACM*, Volume 15, No. 4, April 1972, p. 274.

8. Cox, H. L., Brown A. F. C. Stresses round pins in holes. *The Aeronautical Quarterly*, Vol. XV, pp. 357–372, November 1964.

9. Pilkey, W. D., Pilkey, D. F. Peterson's Stress Concentration Factors. Third Edition, John Wiley & Sons, Inc., 2008, ISBN: 978-0-470-04824-5.

10. Pilkey, W. D. Formulas for stress, strain and structural matrices. John Wiley & Sons, Inc., 1994.

11. Howland, R. C. J. On the stresses in the neighbourhood of a circular hole in a strip under tension. *Philosophical Transactions of the Royal Society of London*. Series A, Containing Papers of a Mathematical or Physical Character, Volume 229, pp. 49–86.

12. Folias, E. S., Wang, J.-J. On the three-dimensional stress field around a circular hole in a plate of arbitrary thickness. *Computational Mechanics*, Vol. 6, 1990, pp. 379–391.

13. Sternberg, E., Sadowsky, M. A. Three-dimensional solution for the stress concentration around a circular hole in a plate of arbitrary thickness. *Transactions of the ASME, Journal of Applied Mechanics*, Vol. 71, March 1949, pp. 27–38.

14. Evans, R. L. Effect of plate thickness on the in-plane and through-thickness stresses at a hole. Technical Report DSTO-TR-0330, Airframes and Engines Division, Defence Science and Technology Organisation, Department of Defence, Australia, April 1996.

15. Penado, F. E., Folias, E. S. The three-dimensional stress field around a cylindrical inclusion in a plate of arbitrary thickness. *International Journal of Fracture*, Vol. 39, 1989, pp. 129–146.

16. Shivakumar, K. N., Newman Jr., J. C. Stress concentrations for straight-shank and countersunk holes in plates subjected to tension, bending and pin loading. NASA Technical Paper 3192, June 1992.

17. Conway, H. D., Vogel, S. M., Farnham, S. M., So, S. Normal and shearing contact stresses in indented strips and slabs. *International Journal of Engineering Science*, Vol. 4, No. 4, 1966, pp. 343–359.

18. Conway, H. D., Farnham, K. A. The contact stress problem for indented strips and slabs under conditions of partial slipping. *International Journal of Engineering Science*, Vol. 5, No. 2, 1967, pp. 145–154.

19. Dundurs, J., Lee, M. S. Stress concentration at a sharp edge in contact problems. *Journal of Elasticity*, Vol. 2, No. 2, 1972, pp. 109–112.

20. Gdoutos, E. E., Theocaris, P. S. Stress concentrations at the apex of a plane indenter acting on an elastic half plane. *Transactions of the ASME, Journal of Applied Mechanics*, Vol. 42, September 1975, pp. 688–692.

21. Comninou, M. Stress singularity at a sharp edge in contact problems with friction. *Journal of Applied Mathematics and Physics* (ZAMP), Vol. 27, No. 4, 1976, pp. 493–499.

22. Miniatt, E., Waas, A., Anderson, W. J. An experimental study of stress singularities at a sharp corner in a contact problem. *Experimental Mechanics*, Vol. 30, No. 3, September 1990, pp. 281–285.

23. Bijak-Zochowski, M., Waas, A., Anderson, W., Miniatt, C. Reduction of contact stress by use of relief notches. *Experimental Mechanics*, Vol. 31, No. 3, 1991, pp. 271–275.

24. Bijak-Zochowski, M., Marek, P., Tracz, P. On methods of reduction and elimination of stress singularities in some elastic contact problems. *International Journal of Mechanical Sciences*, Vol. 36, No. 4, 1994, pp. 279–296.

25. Bijak-Zochowski, M., Marek, P. Development of plastic zones and residual stress in elasto-plastic contact problems with stress singularities in elastic range. *International Journal of Mechanical Sciences*, Vol. 38, No. 2, 1996, pp. 175–190.

26. Sinclair, G. B., Cormier, N. G., Griffin, J. H., Meda, G. Contact stresses in dovetail attachments: finite element modelling. *Journal of Engineering for Gas Turbines and Power*, Vol. 124, No. 1, 2002, pp. 182–189.

27. Grant, R. J., Flipo, B. C. D. Parametric study of the elastic stress distribution in pin-loaded lugs modelled in two and three dimensions and loaded in tension. *The Journal of Strain Analysis for Engineering Design*, Vol. 44, No. 6, 2009, pp. 473–489.

28. Whitcomb, J. D., Raju, I. S., Goree, J. G. Reliability of the finite element method for calculating free edge stresses in composite laminates. *Computers & Structures*, Vol. 15, No. 1, 1982, pp. 23–37.

Table 1: *Comparison of the peak tensile and peak compressive K_t values for a hole in a uniaxially-loaded plate obtained using the 2D analytical solution for an infinite plate (W/D = ∞) and 2D FEA of a square finite plate with W/D = 40 with two levels of mesh refinement.*

	Analytical Solution Infinite Plate $W/D = \infty$	FEA With Coarse Mesh (20 equispaced elements around hole boundary) Square Finite Plate $W/D = 40$		FEA With Fine Mesh (180 equispaced elements around hole boundary) Square Finite Plate $W/D = 40$	
	K_t	K_t	Error	K_t	Error
Tensile peak	3.00000	3.01415	+0.472%	3.00539	+0.180%
Compressive peak	1.00000	1.01009	+1.009%	1.00425	+0.425%

Table 2: *Radial and tangential stresses at selected angles around the hole–pin contact interface obtained from 2D FEA and the finite-width-corrected 2D analytical infinite-plate solution for an aluminium coupon and titanium pin.*

	2D FEA of Coupon		2D Analytical Infinite-Plate Solution With Finite-Width Correction
	Coarse Mesh	Fine Mesh	
η	20.25°	19.75°	19.31°
σ_r/S at θ = 0°	0.7778	0.7836	0.7766
σ_t/S at θ = 0°	2.6228	2.6190	2.6599
σ_t/S at θ = η	2.8628	2.9197	2.9731
σ_t/S at θ = 90°	–0.8001	–0.7978	–0.8015

Table 3: *Radial and tangential stresses at selected angles around the hole–pin contact interface at the midplane (z/t = 0) and surface (z/t = 0.5) faces of the coupon for the aluminium coupon and titanium pin using 3D FEA with different levels of mesh refinement.*

	Fine Mesh		Coarse Mesh		Graded Mesh	
	Midplane	Surface	Midplane	Surface	Midplane	Surface
η	19.00°	22.50°	20.08°	22.29°	20.50°	22.79°
σ_r/S at θ = 0°	0.7757	1.7033	0.7558	1.3114	0.7720	1.5986
σ_t/S at θ = 0°	2.7005	2.3173	2.7164	2.4433	2.7042	2.3534
σ_t/S at θ = η	3.0310	2.6120	2.9668	2.5494	2.9543	2.5514
σ_t/S at θ = 90°	–0.9203	–0.4917	–0.9304	–0.4887	–0.9222	–0.4918

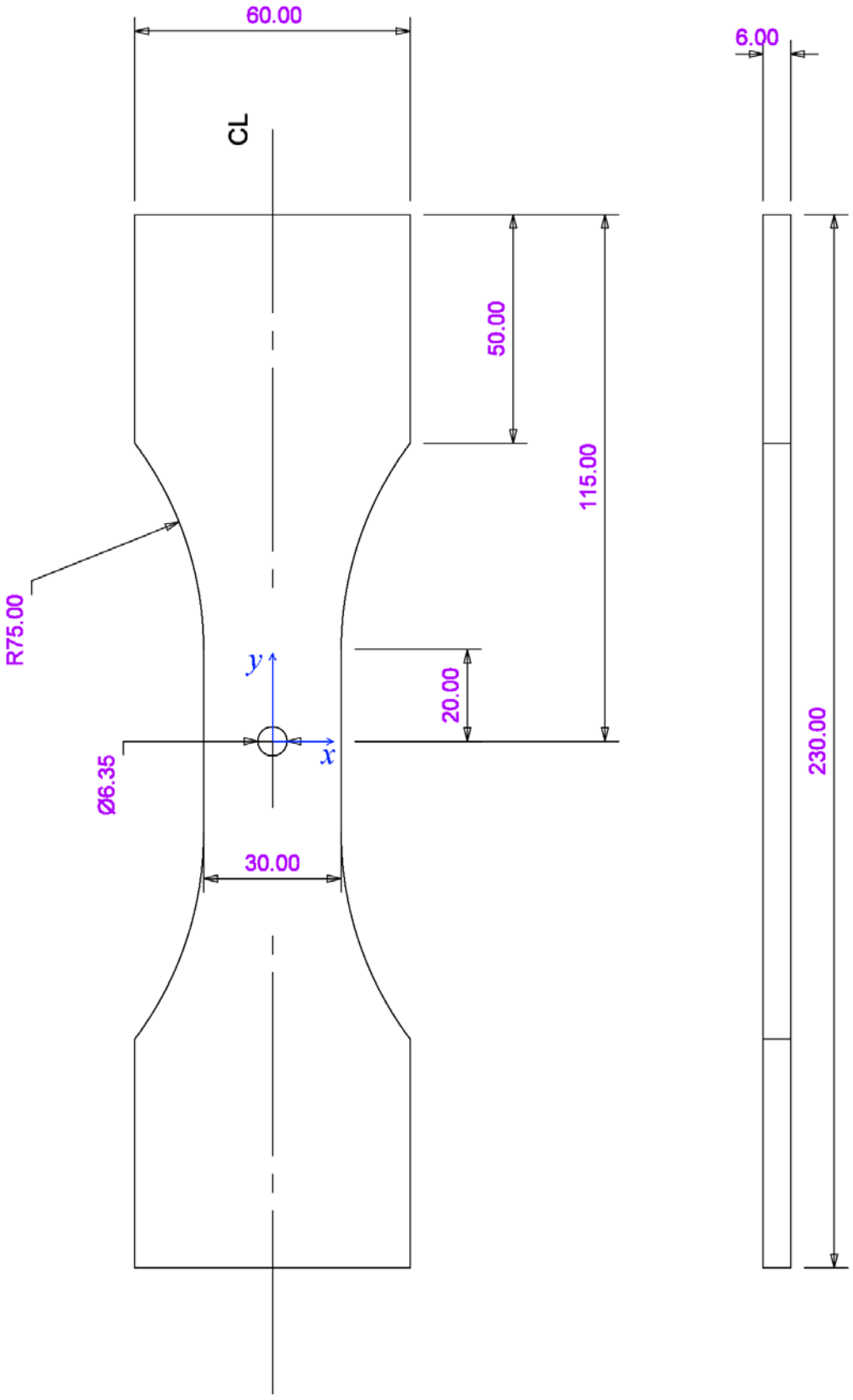


Figure 1: *Drawing showing dimensions of LIF Hawk Filled Hole Coupon to be used in fatigue testing, including the location of the origin of the* x-y *coordinate system.*

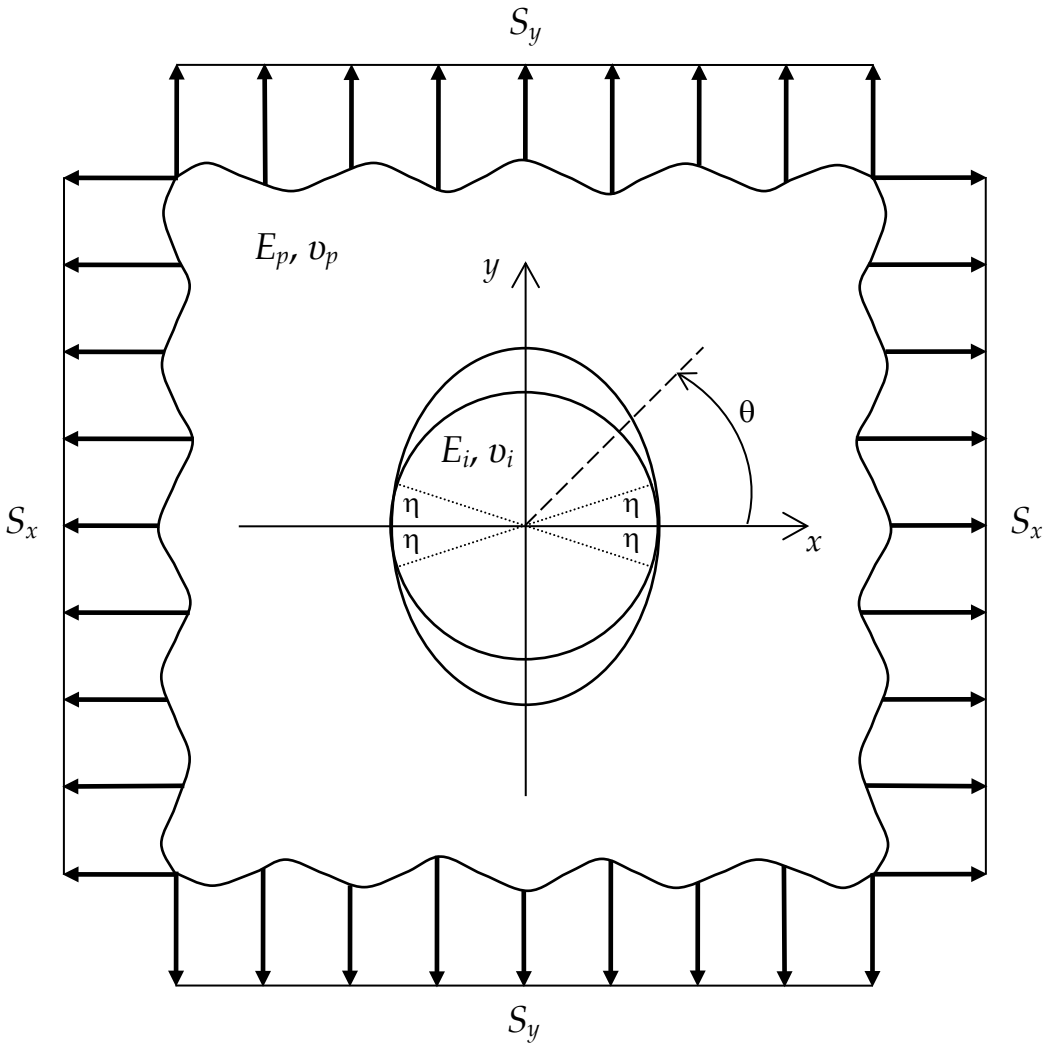
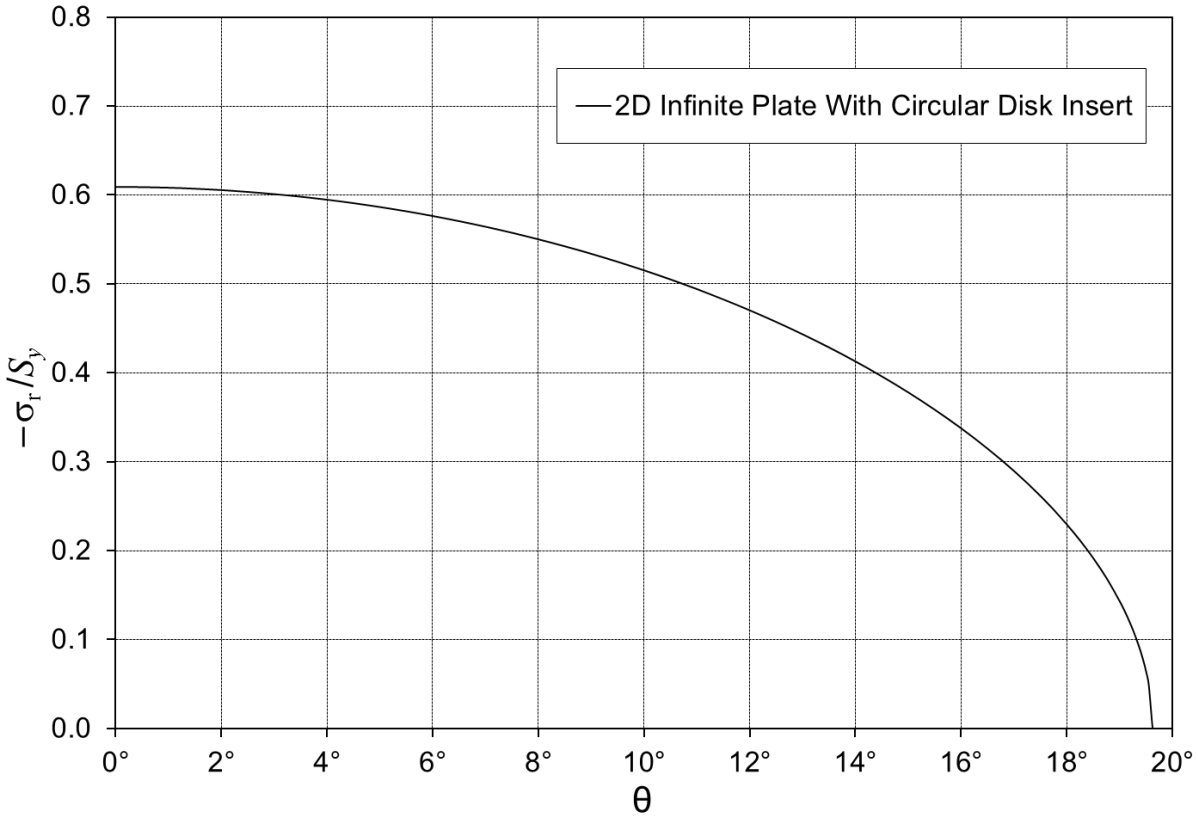
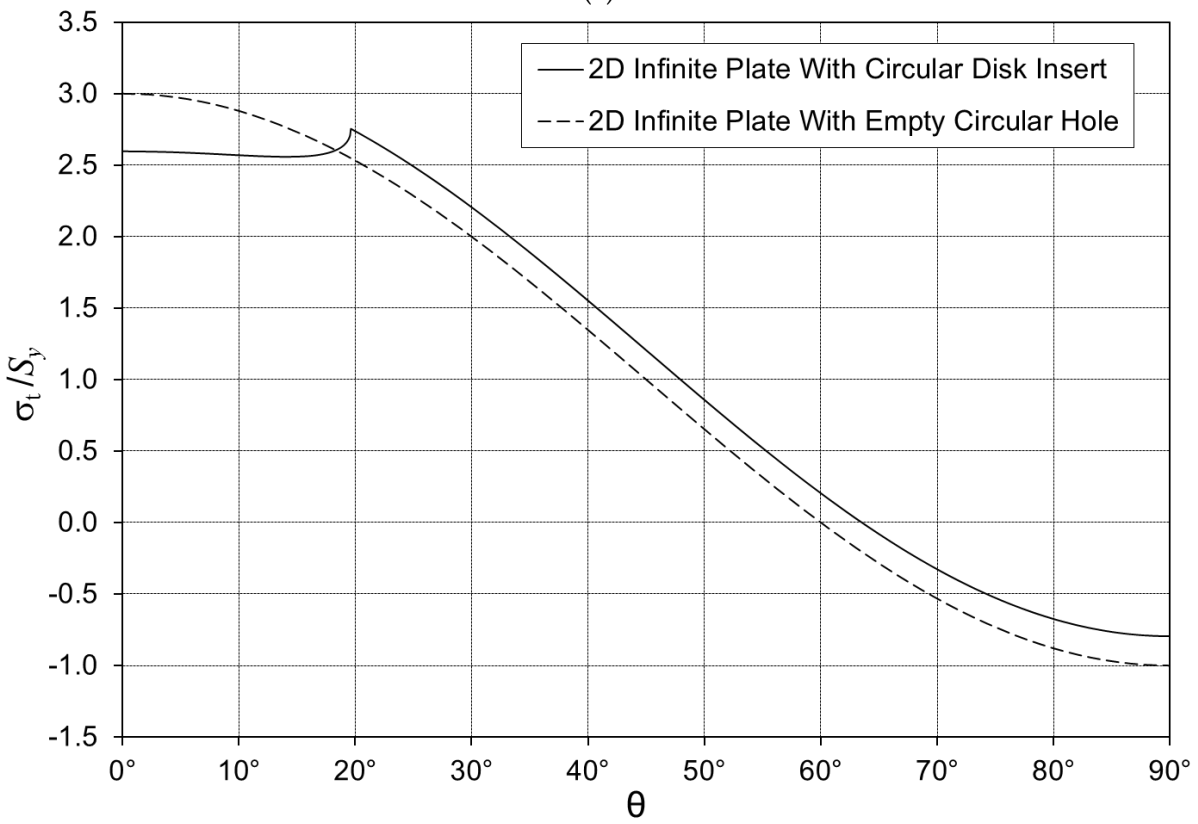


Figure 2: *Geometrical configuration of an infinite 2D plate loaded by stresses* S_x *and* S_y *at infinity showing the contact angle* η *between the plate and the circular disk insert.*

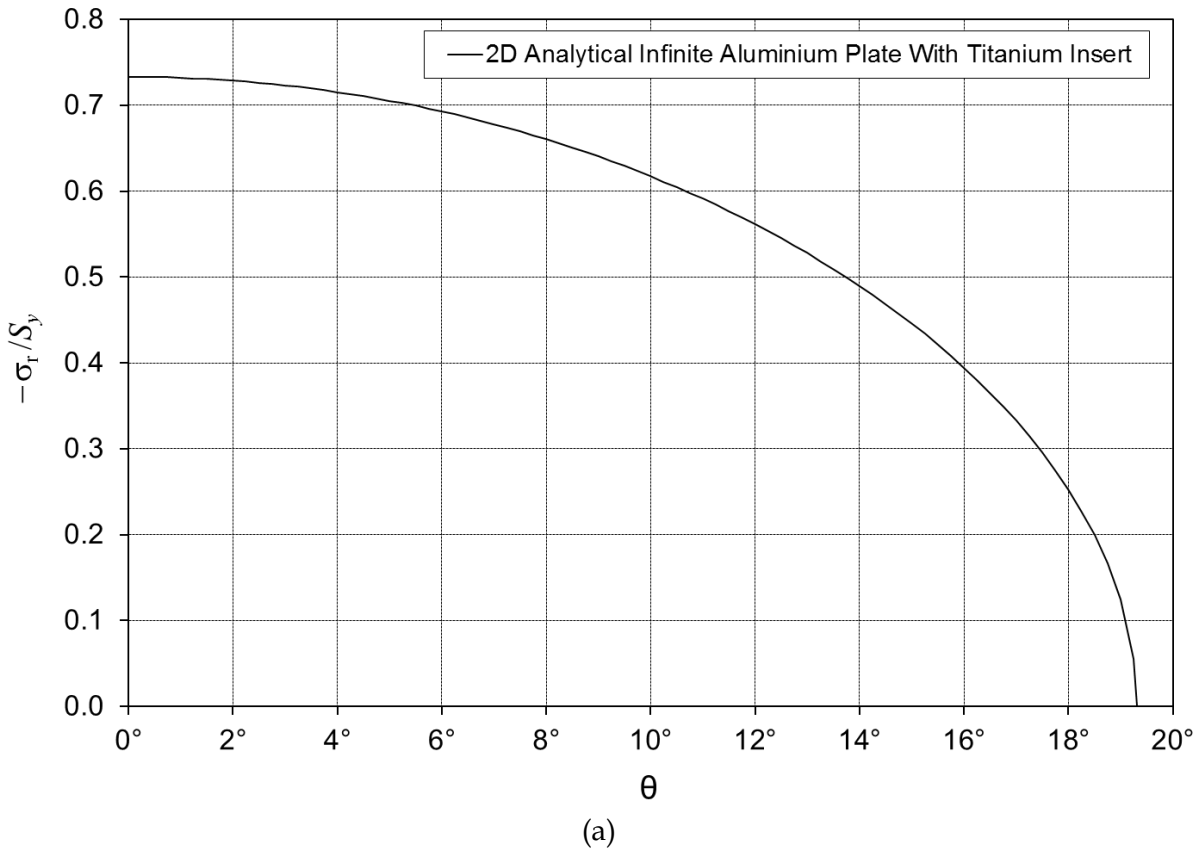


(a)

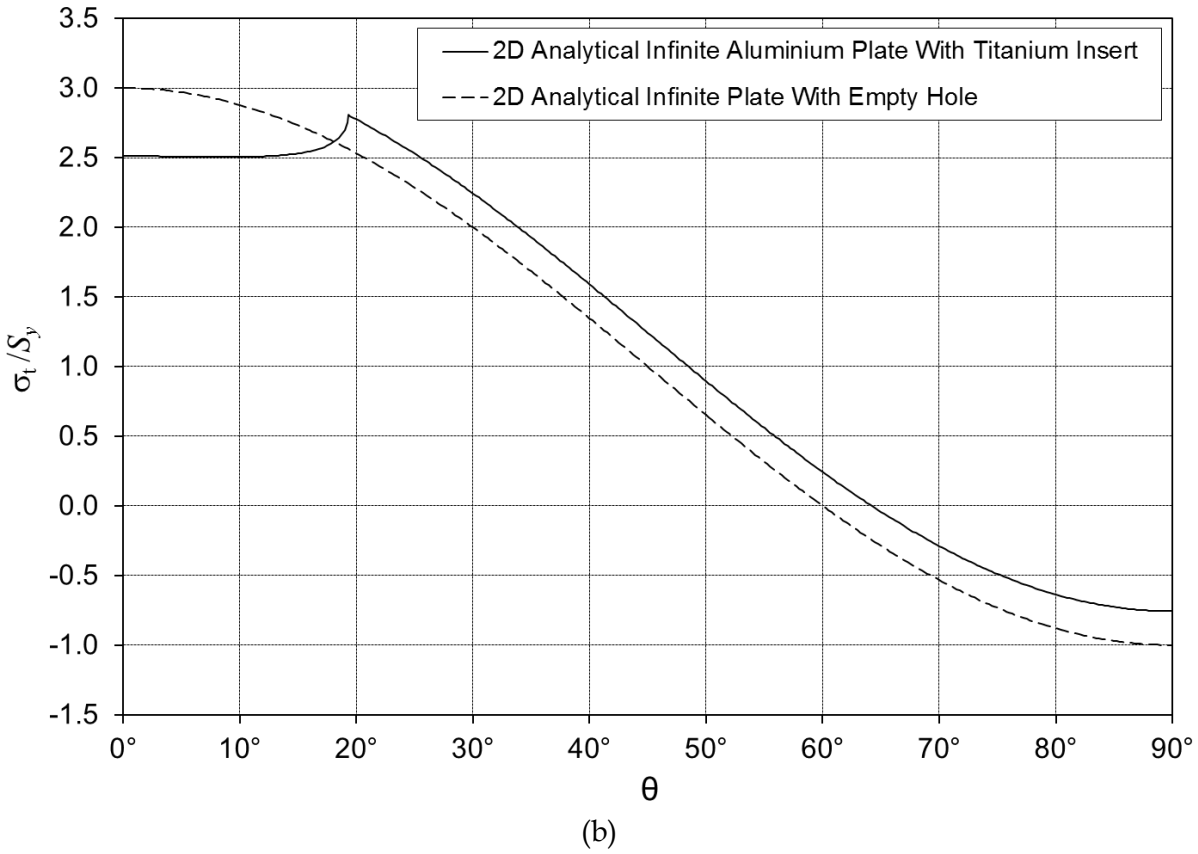


(b)

Figure 3: *Stress distribution around boundary of a circular hole with a circular disk insert of the same material, in a 2D infinite uniaxially-loaded plate. (a) Radial stress. (b) Tangential stress. The stress around an empty circular hole in an infinite plate is also shown for comparison.*



(a)



(b)

Figure 4: *Stress distribution around the boundary of a circular hole with a titanium circular disk insert, in a 2D infinite uniaxially-loaded aluminium plate. (a) Radial stress. (b) Tangential stress. Stresses for an infinite plate with an empty circular hole also shown for comparison.*

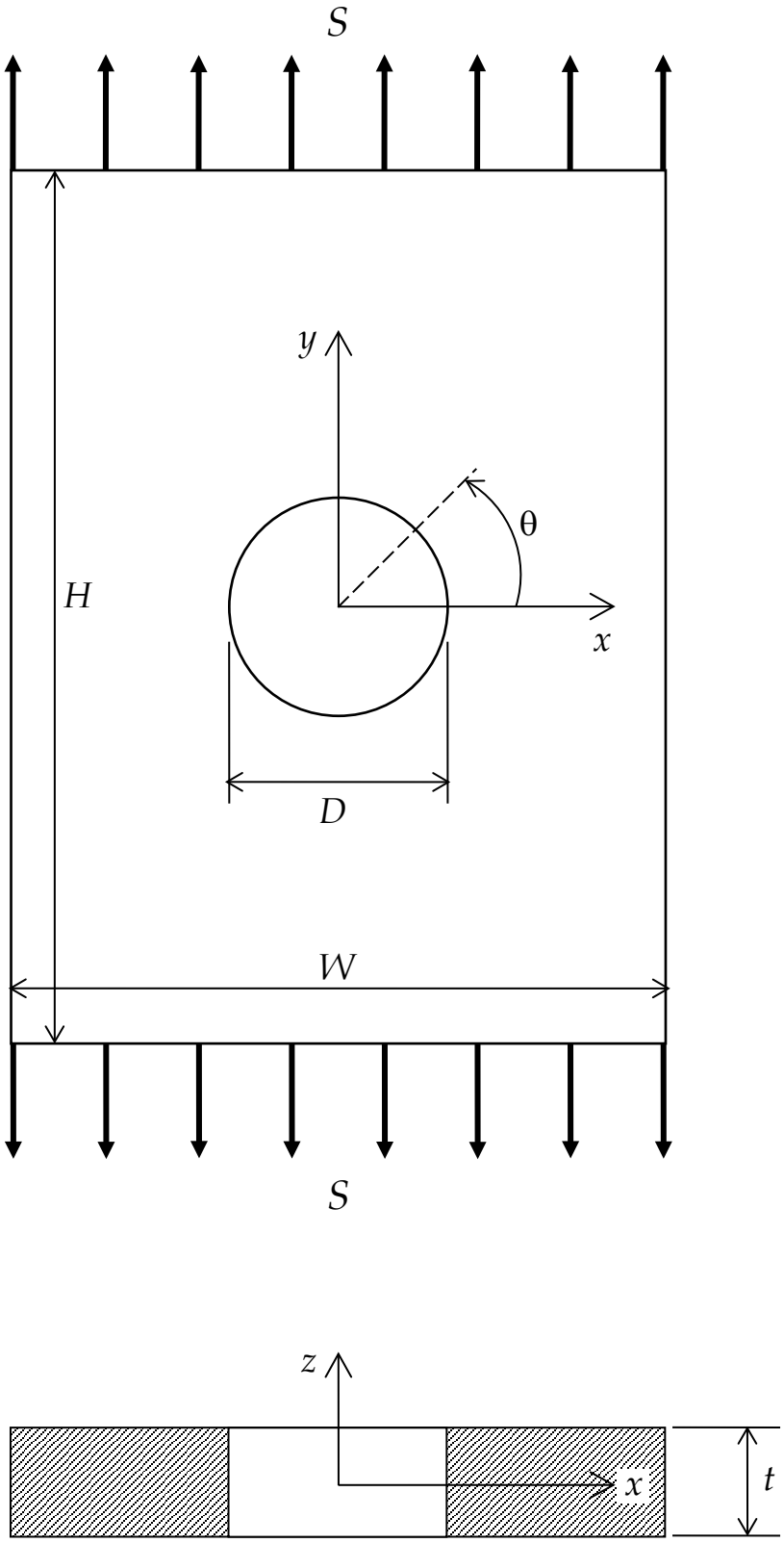
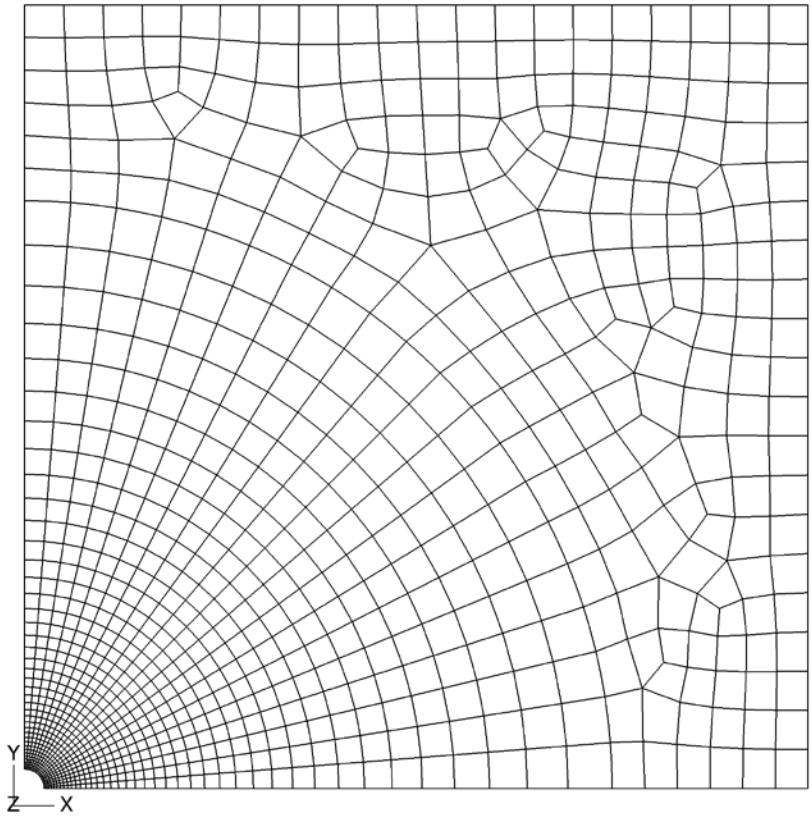
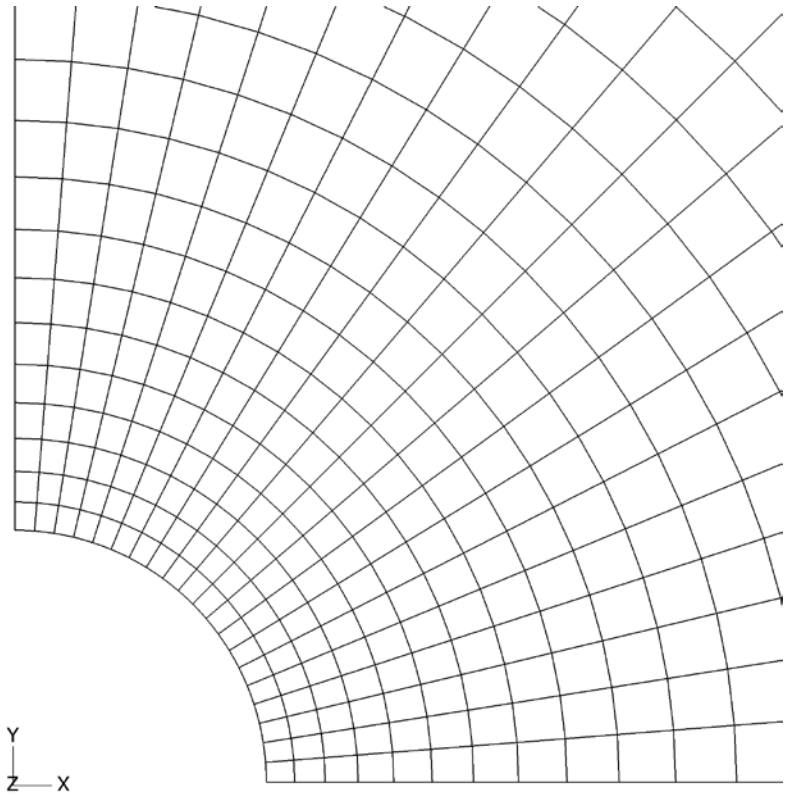


Figure 5: *Geometrical configuration of a 3D plate of height* H *and width* W *and thickness* t *with a central circular hole of diameter* D.

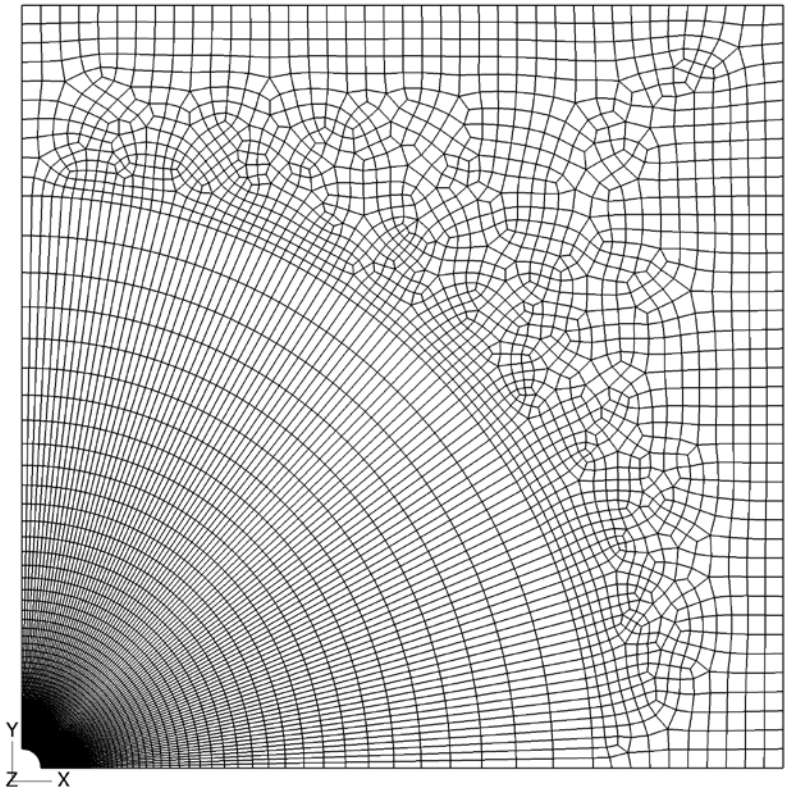


(a)

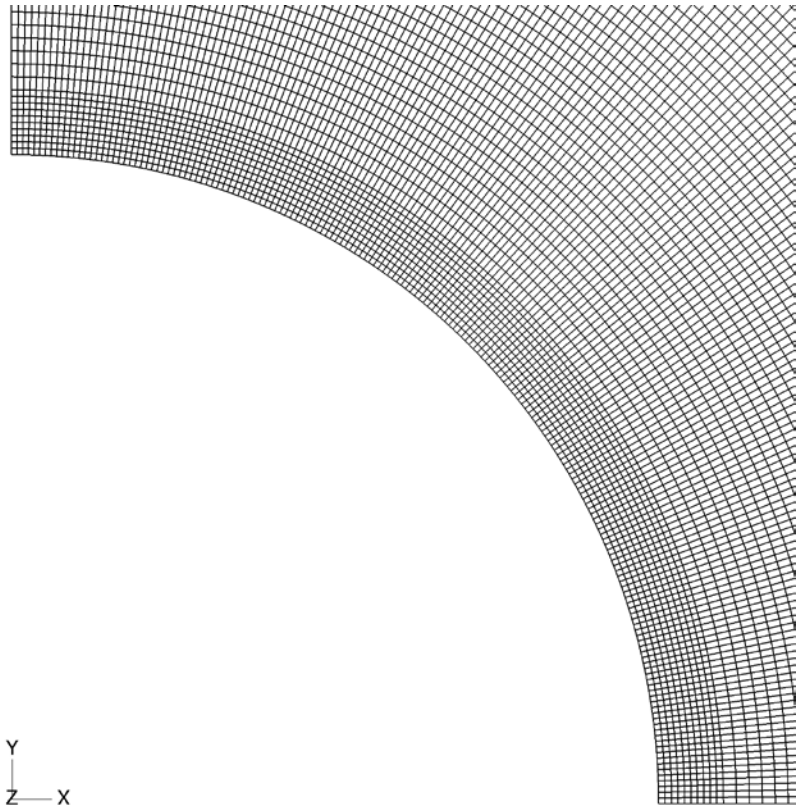


(b)

Figure 6: *2D ¼-symmetry Abaqus coarse-mesh finite element model of a uniaxially-loaded square plate with a circular hole (W/D = 40), with 20 elements around ¼-circumference of hole. (a) Entire plate. (b) Detail of mesh in vicinity of hole.*



(a)



(b)

Figure 7: *2D ¼-symmetry Abaqus fine-mesh finite element model of a uniaxially-loaded square plate with a circular hole (W/D = 40), with 180 elements around ¼-circumference of hole. (a) Entire plate. (b) Detail of mesh in vicinity of hole.*

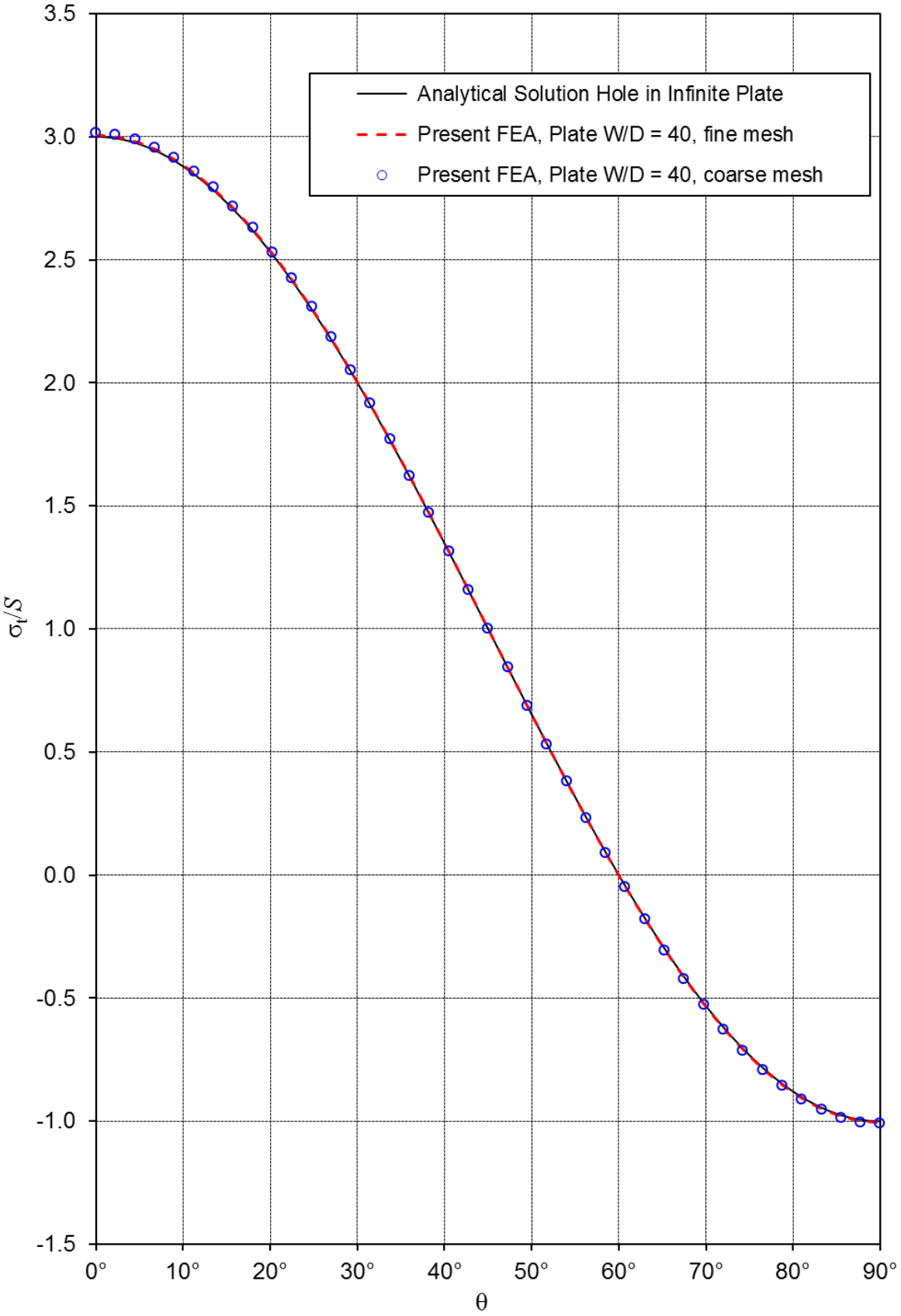
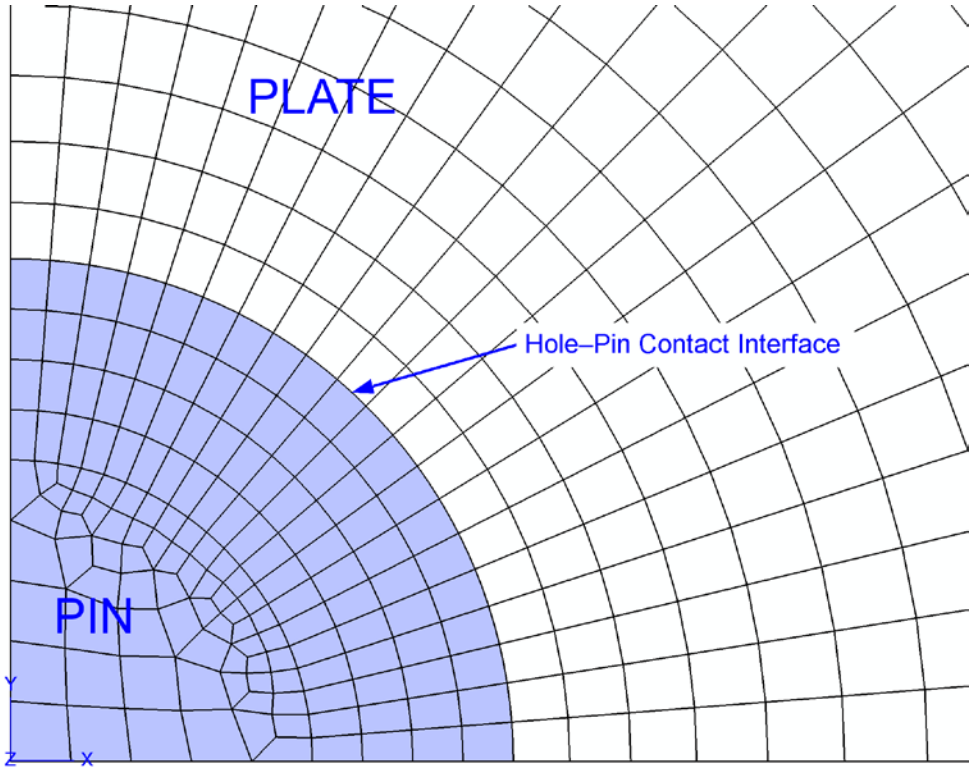
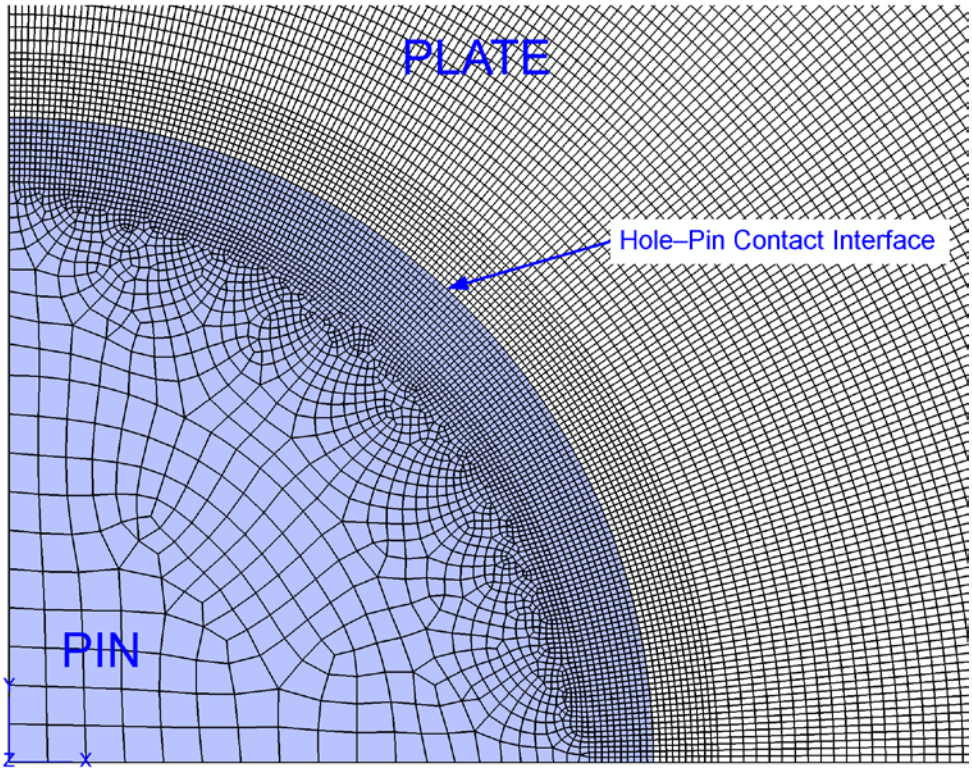


Figure 8: *Comparison between the analytical and FEA solutions for the distribution of normalised tangential stress around the boundary of a hole in a uniaxially-loaded square plate. Coarse-mesh FEA case corresponds to 20 equispaced elements around the ¼-circumference of the hole boundary, while fine-mesh FEA case uses 180 equispaced elements.*

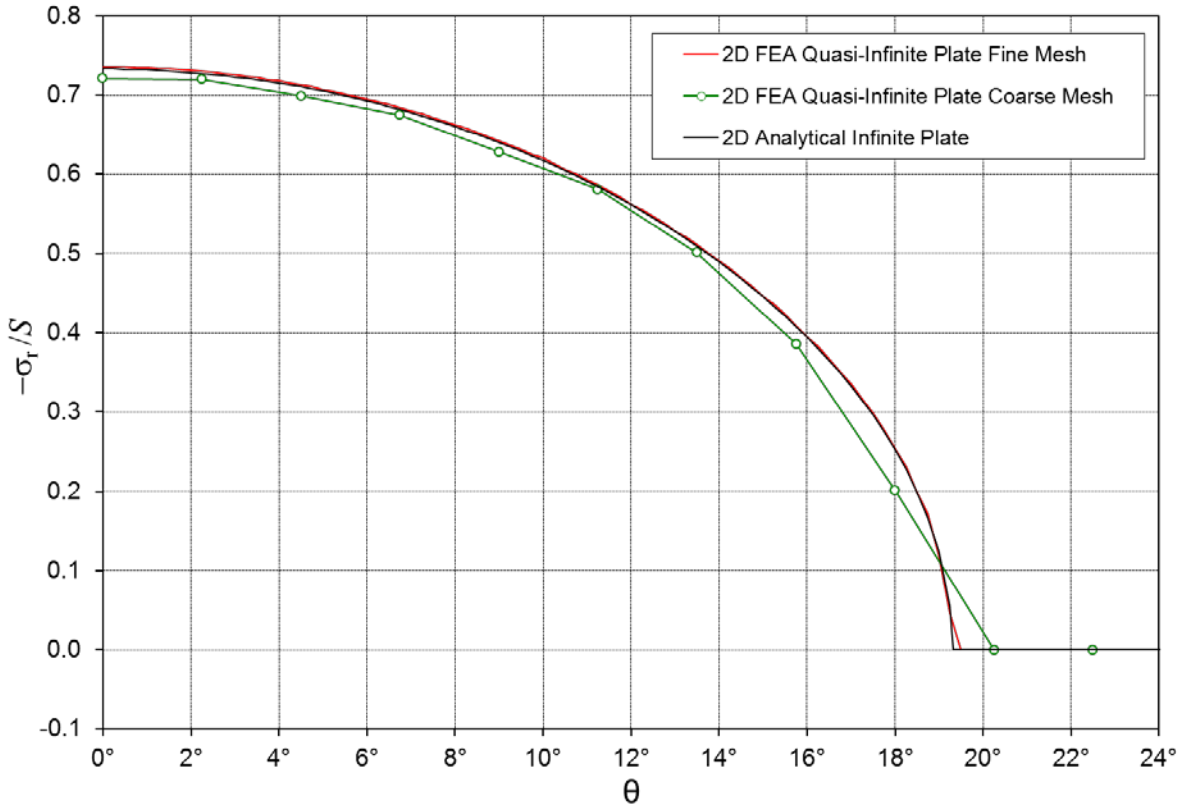


(a)

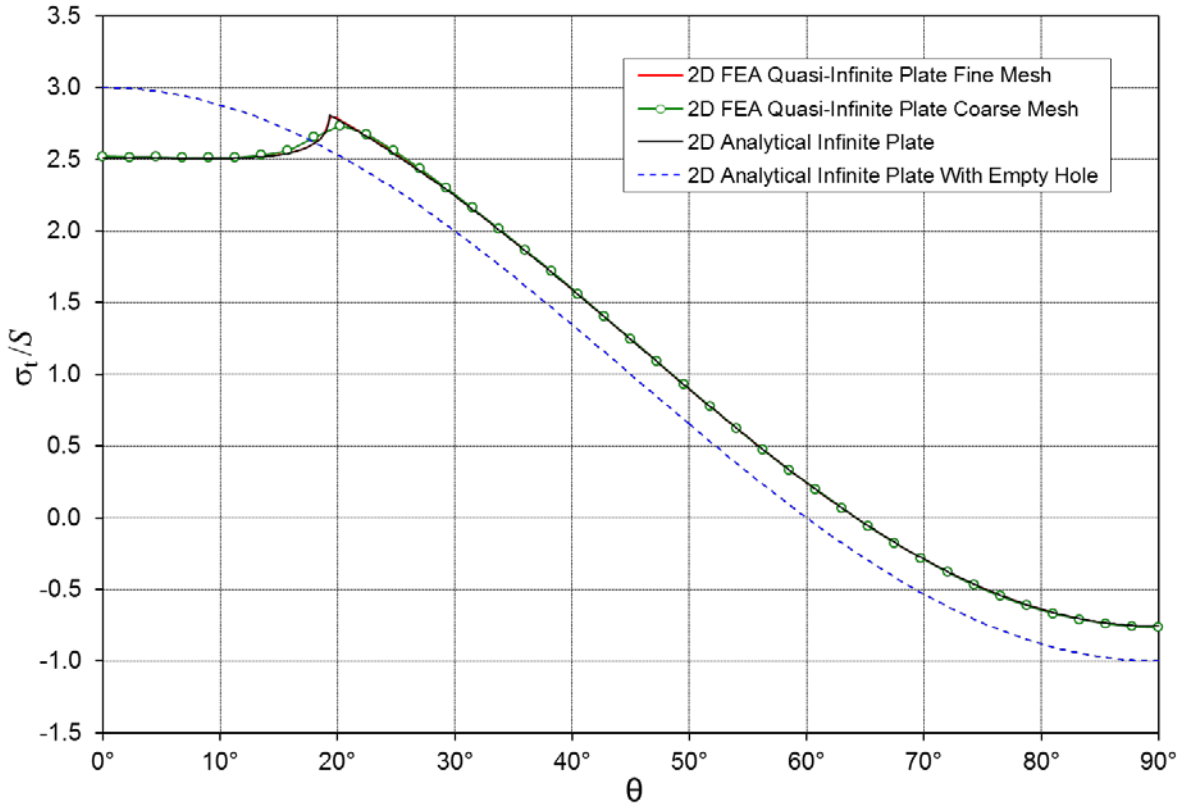


(b)

Figure 9: *Details of 2D ¼-symmetry Abaqus models of a uniaxially-loaded square plate with a circular hole (W/D = 40) and a titanium pin in vicinity of the hole–pin contact interface. (a) Coarse-mesh model with 20 equispaced elements around ¼-circumference of hole. (b) Fine-mesh model with 180 equispaced elements around ¼-circumference of hole.*

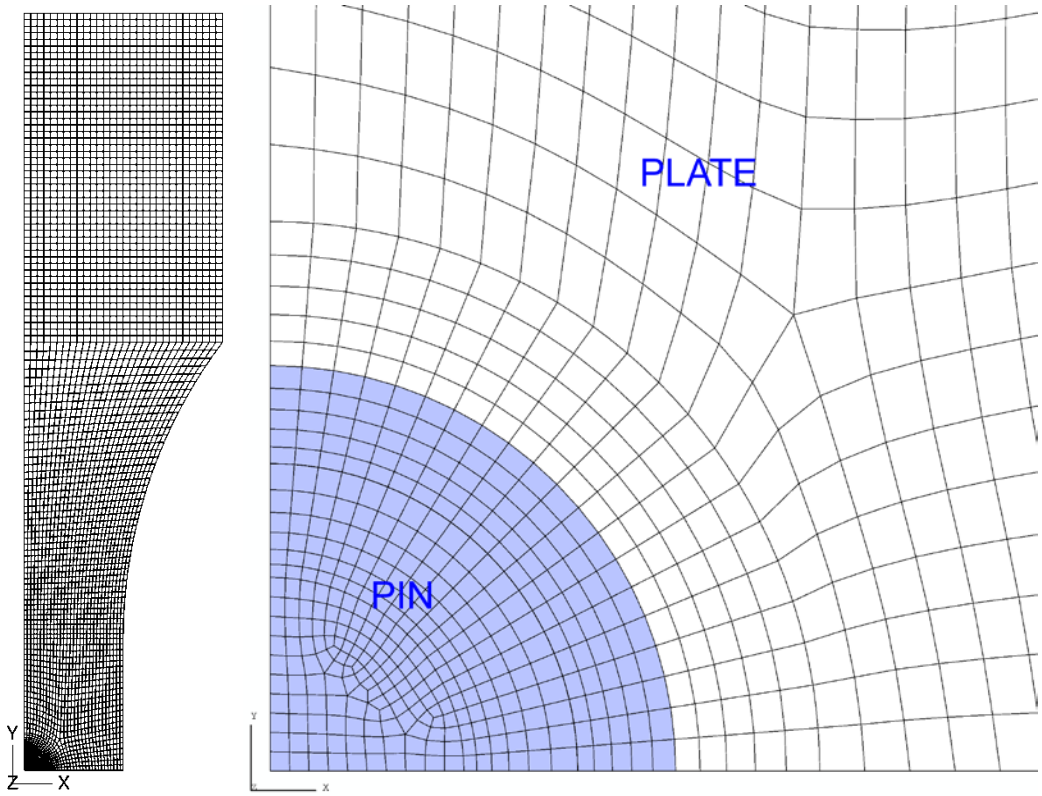


(a)

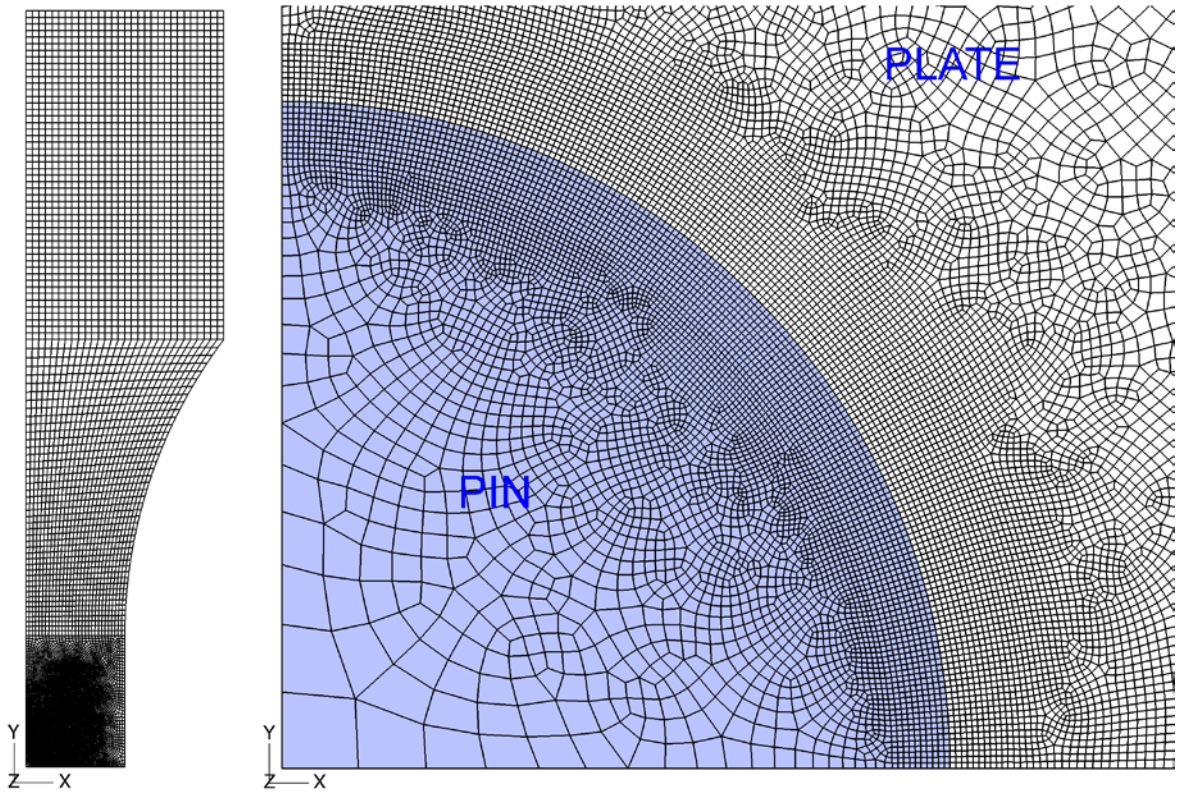


(b)

Figure 10: *Variation of stress around a hole in a uniaxially-loaded square aluminium plate with a neat-fit titanium pin. Results obtained from Abaqus 2D fine-mesh and coarse-mesh FEA for a quasi-infinite plate with* W/D = *40 and an analytical infinite-plate solution. (a) Radial stress. (b) Tangential stress.*

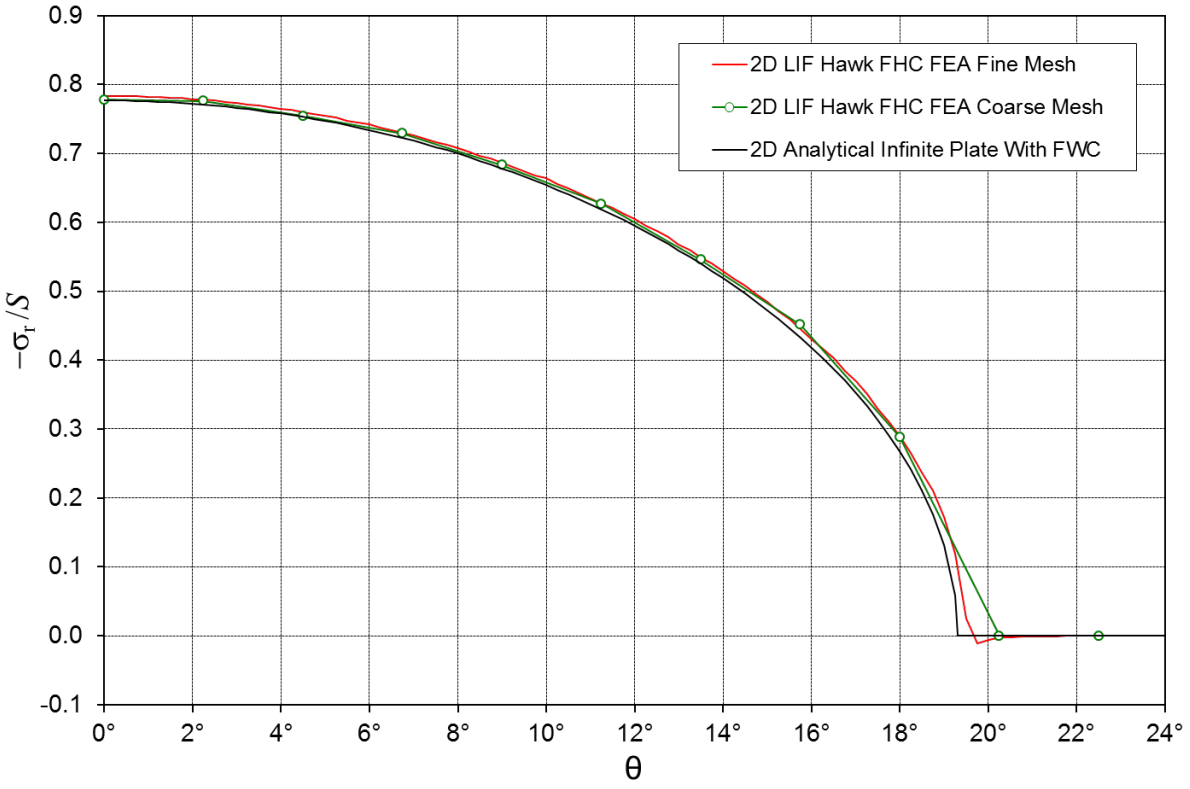


(a)

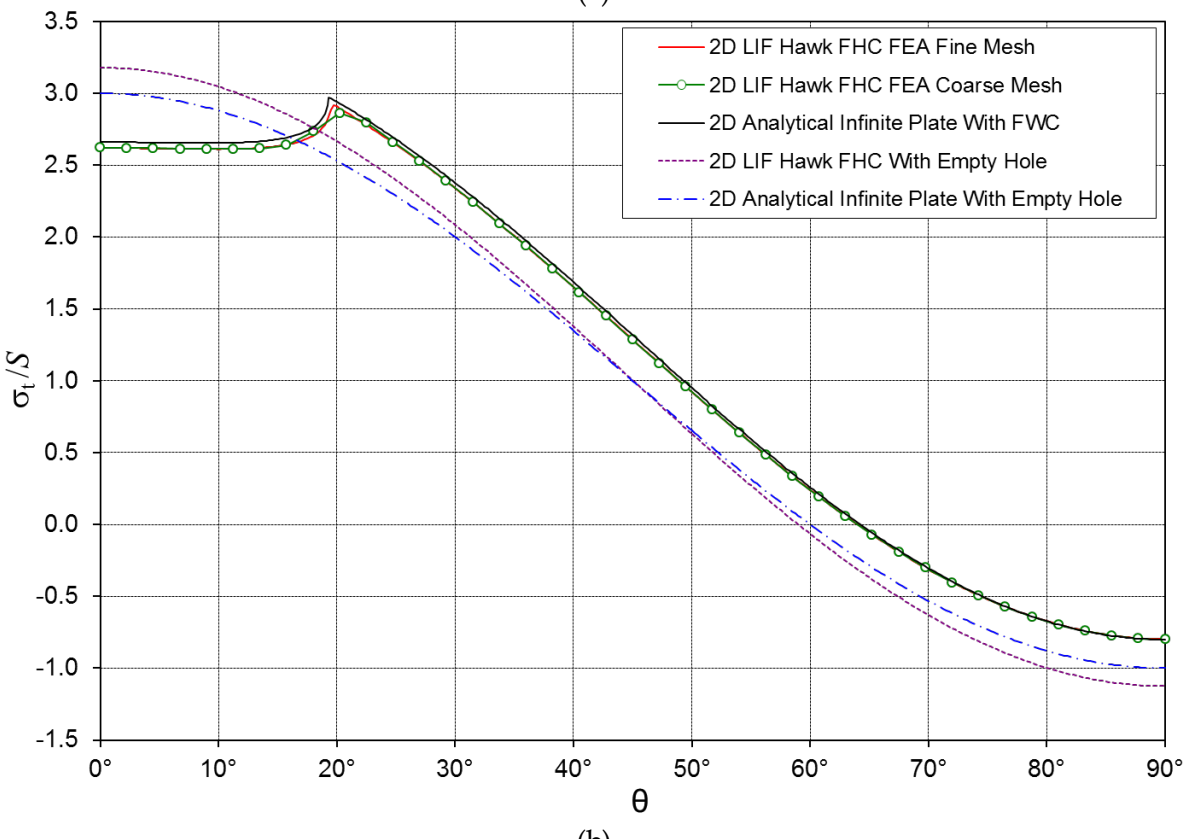


(b)

Figure 11: *Abaqus 2D ¼-symmetry finite element mesh used to model the aluminium coupon and the neat-fit titanium pin. (a) Coarse-mesh model with 20 equispaced elements around the ¼-circumference of the hole–pin boundary. (b) Fine-mesh model with 180 equispaced elements around the ¼-circumference of the hole–pin boundary.*



(a)



(b)

Figure 12: *Variation of stress around a hole in uniaxially-loaded aluminium coupon with a neat-fit titanium pin. Results from Abaqus 2D FEA fine-mesh and coarse-mesh models, as well as the finite-width-corrected analytical infinite-plate solution. (a) Radial stress. (b) Tangential stress.*

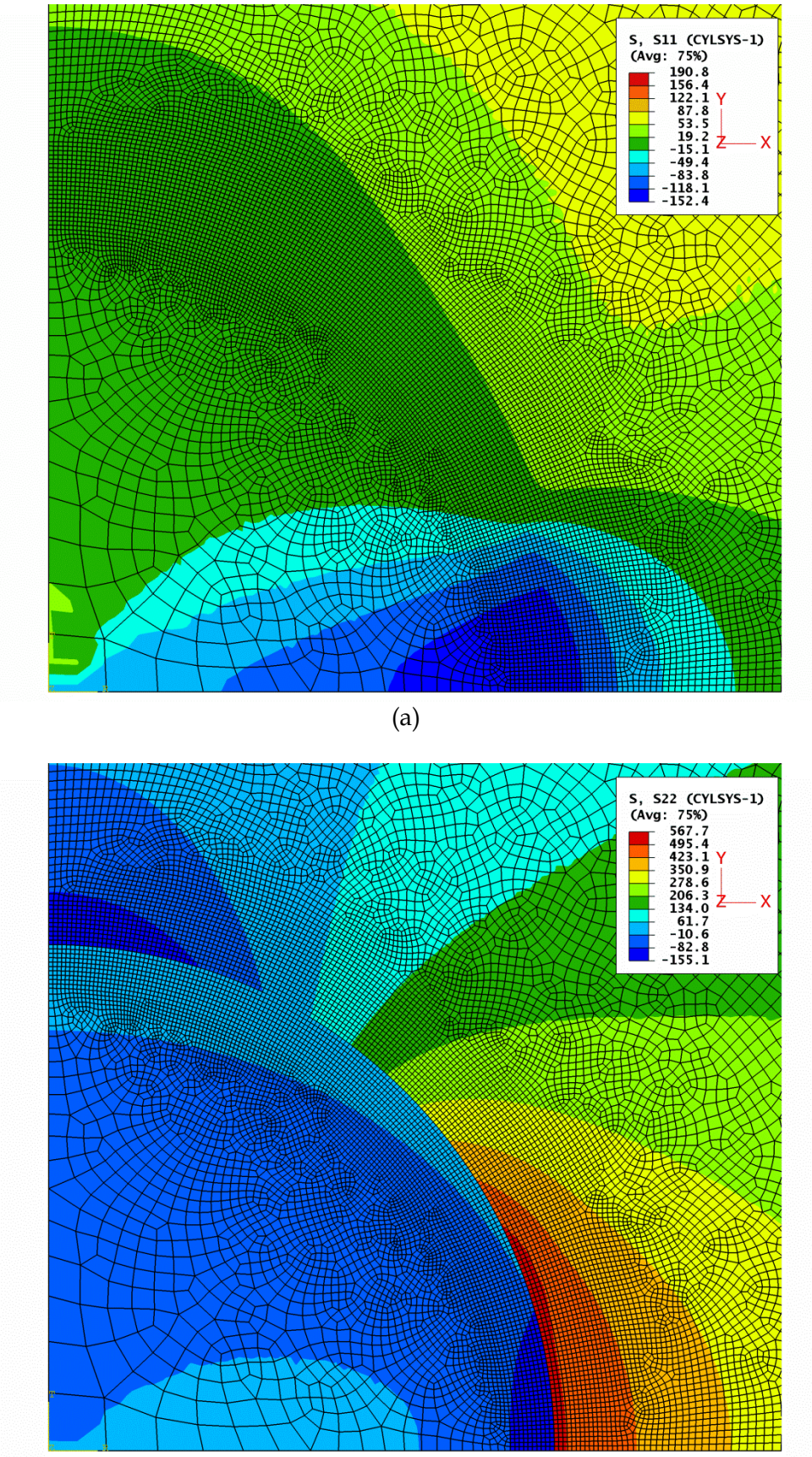


Figure 13: *Stress contours for Abaqus 2D fine-mesh finite element model of aluminium coupon and neat-fit titanium pin. (a) Radial stress. (b) Tangential stress.*

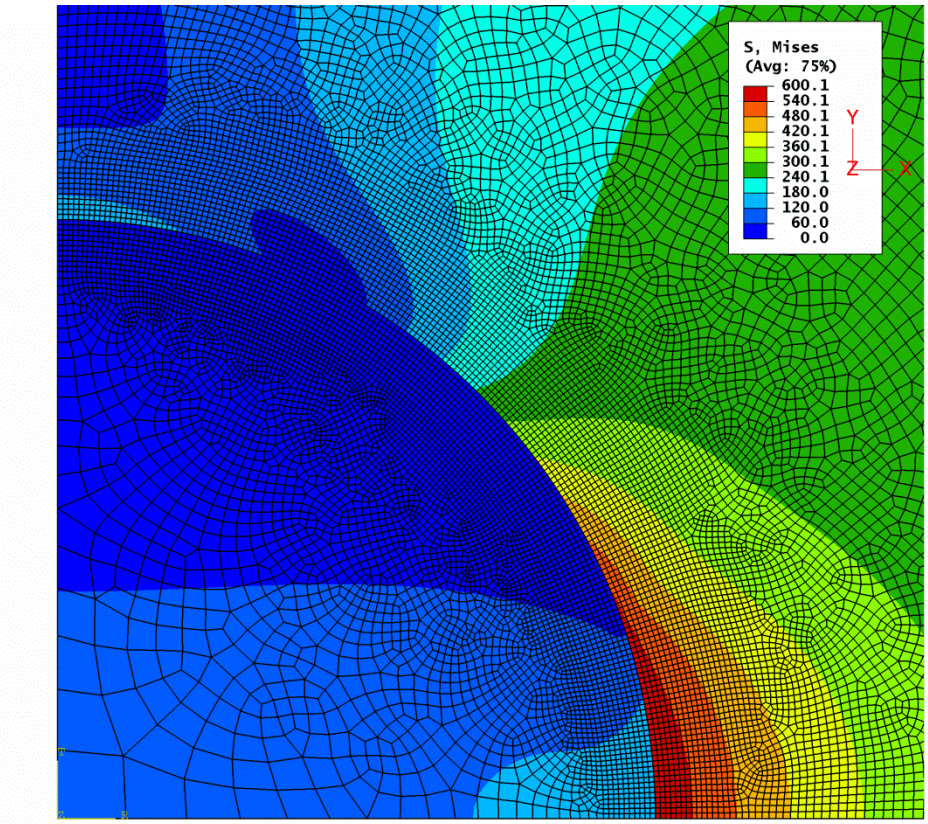
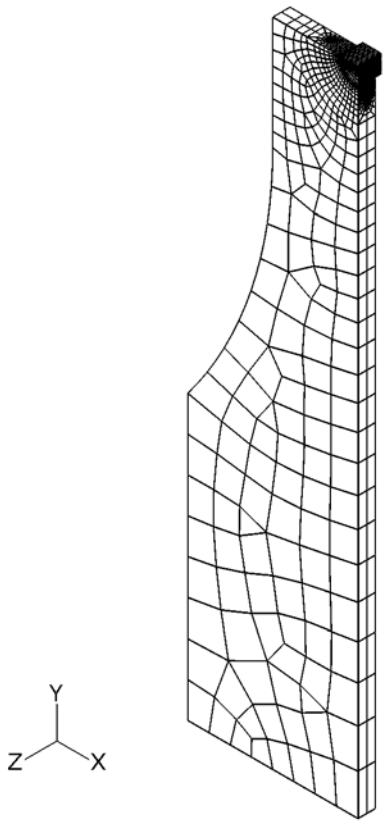
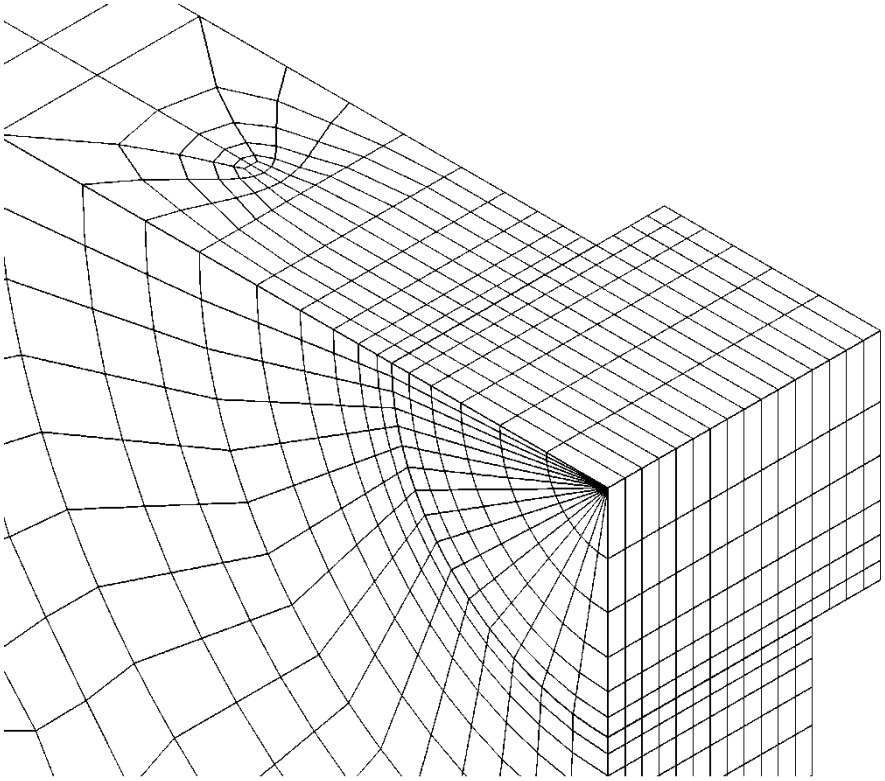


Figure 14: *Von Mises stress contours for Abaqus 2D fine-mesh finite element model of aluminium coupon and neat-fit titanium pin.*

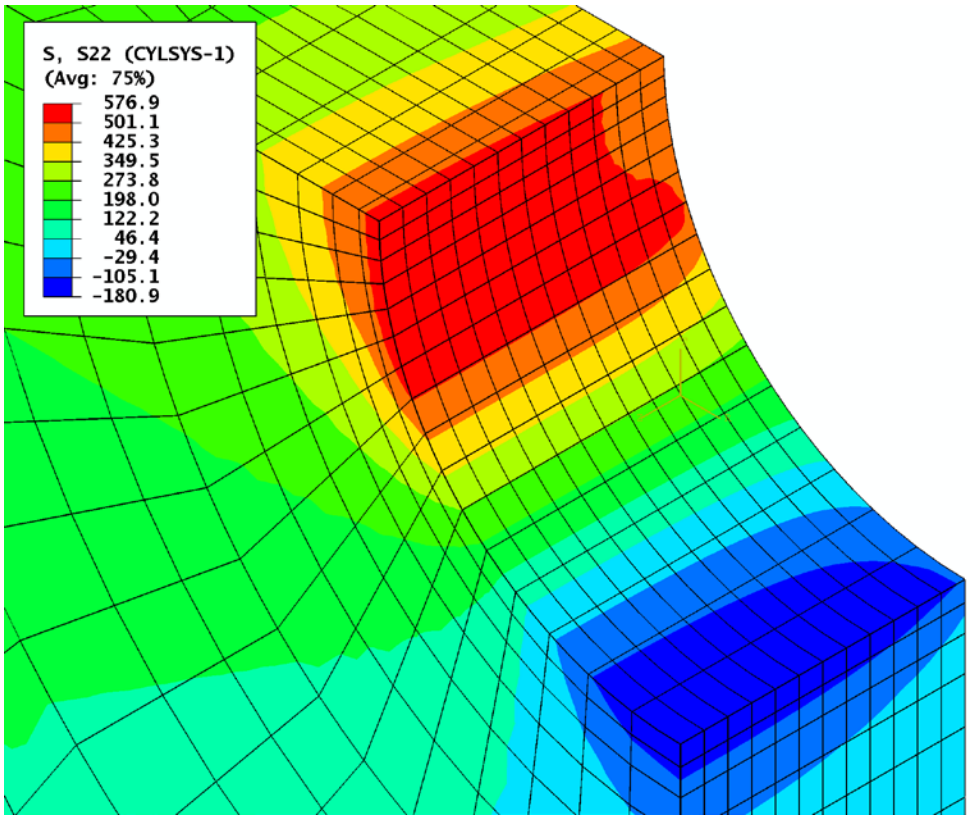


(a)

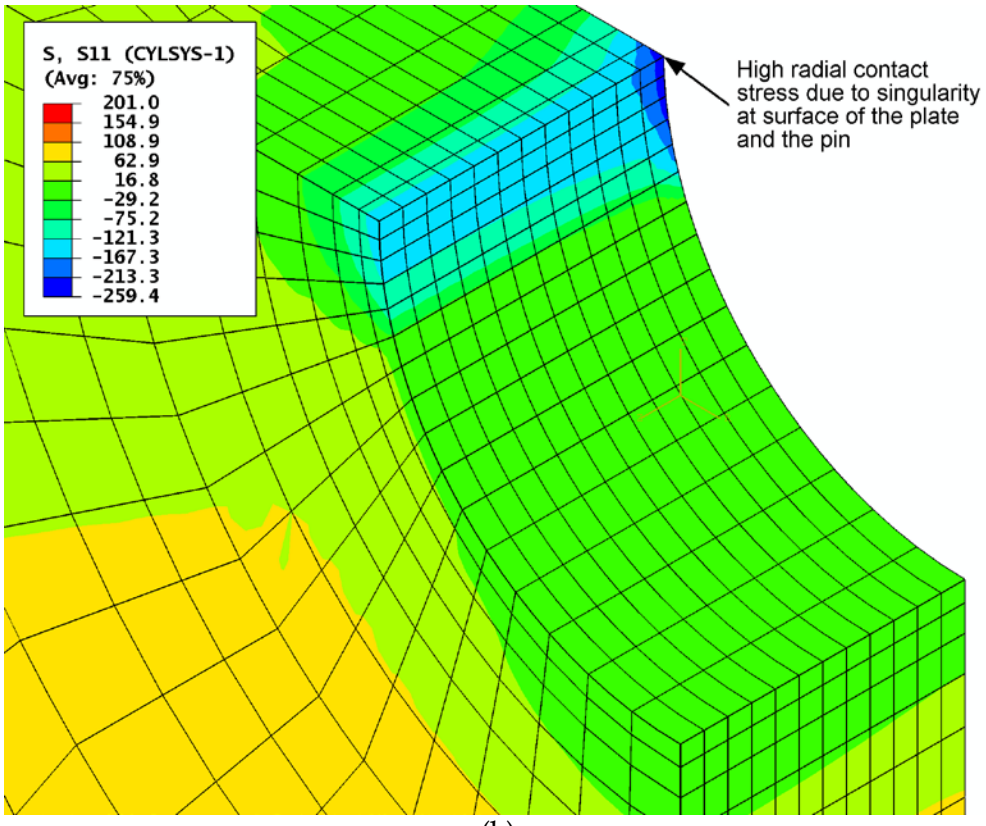


(b)

Figure 15: *Abaqus 3D coarse-mesh finite element model of the aluminium coupon and the neat-fit titanium pin. (a) Complete ⅛-symmetry model (midplane surface is on the left). (b) Detail of finite element mesh in vicinity of hole in the coupon, including the pin.*

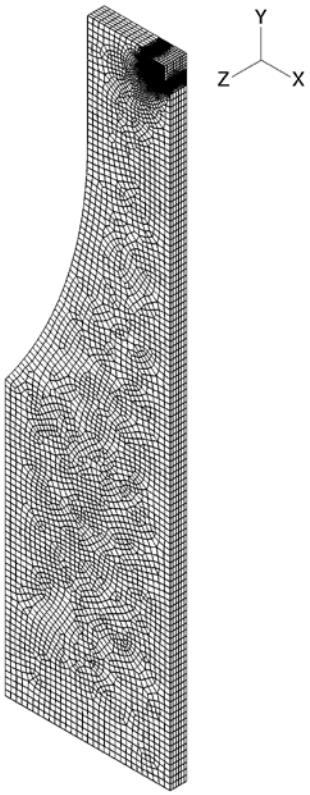


(a)

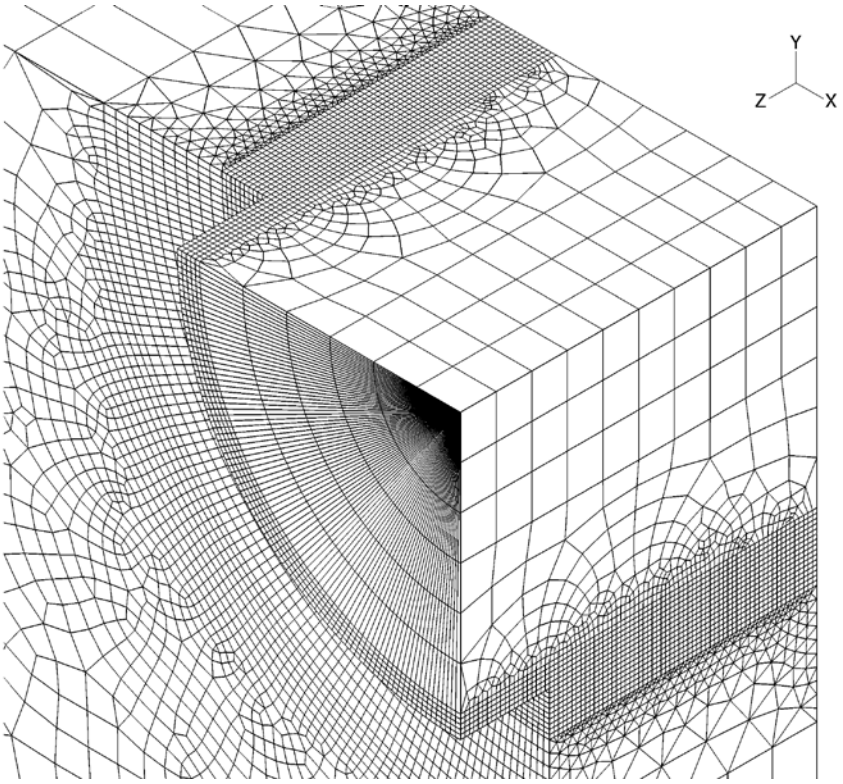


(b)

Figure 16: *Stress contours in the vicinity of the hole for the aluminium coupon with neat-fit titanium pin for the 3D coarse finite element mesh. The midplane surface of the coupon is on the left, and the outer (free) surface is on the right. (a) Tangential stress. (b) Radial stress.*

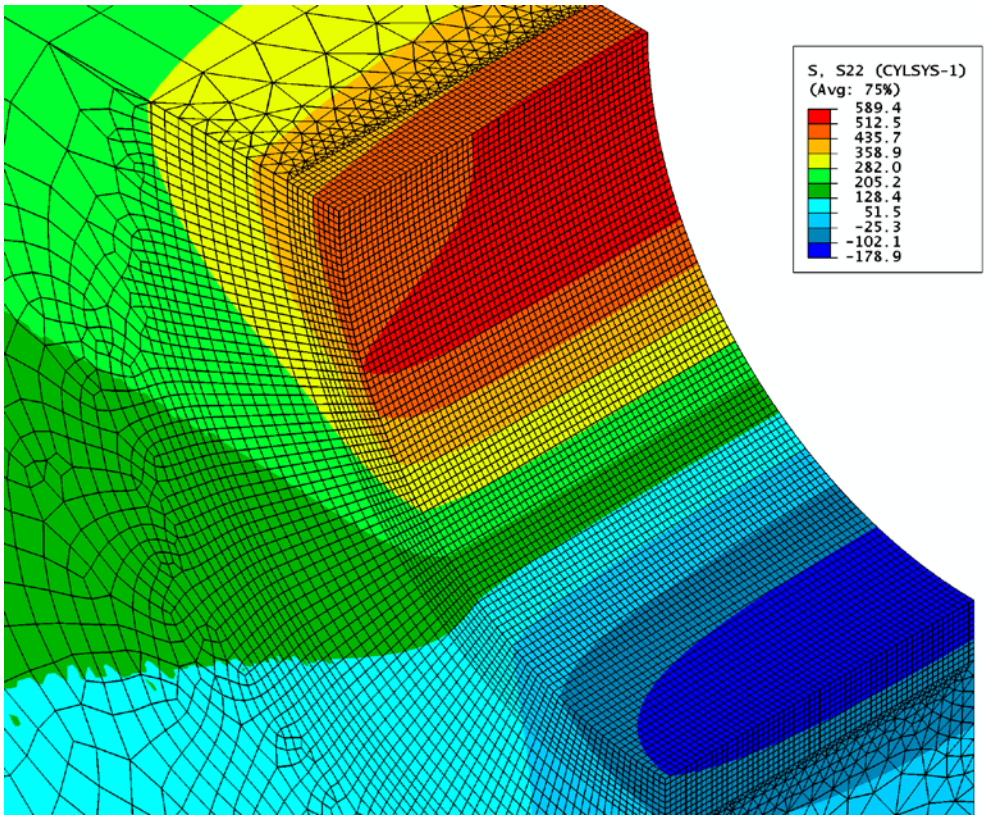


(a)

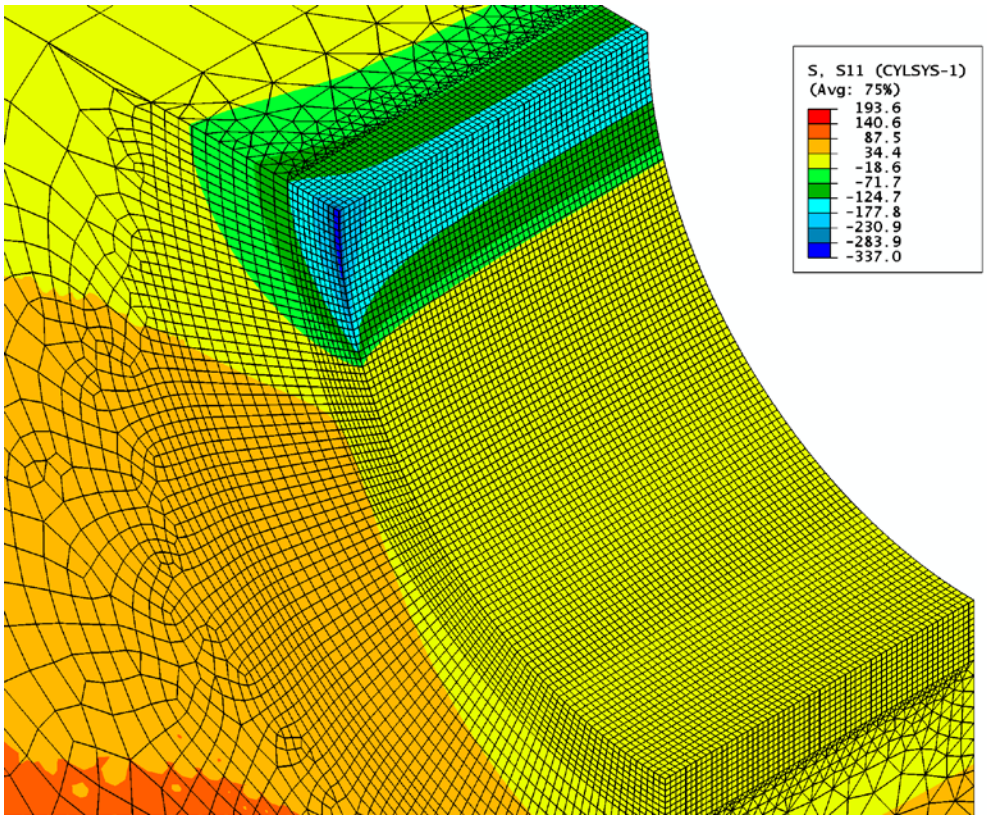


(b)

***Figure* 17:** *Abaqus 3D fine-mesh finite element model of the aluminium coupon and the neat-fit titanium pin, with 90 elements around the ¼-circumference of the hole–pin boundary. (a) Complete ⅛-symmetry model (midplane surface is on the right). (b) Detail of finite element mesh in vicinity of hole in the coupon, including the pin.*



(a)



(b)

Figure 18: *Stress contours in the vicinity of the hole for the aluminium coupon with neat-fit titanium pin for the 3D fine finite element mesh. The midplane surface of the coupon is on the right, and the outer (free) surface is on the left. (a) Tangential stress. (b) Radial stress*

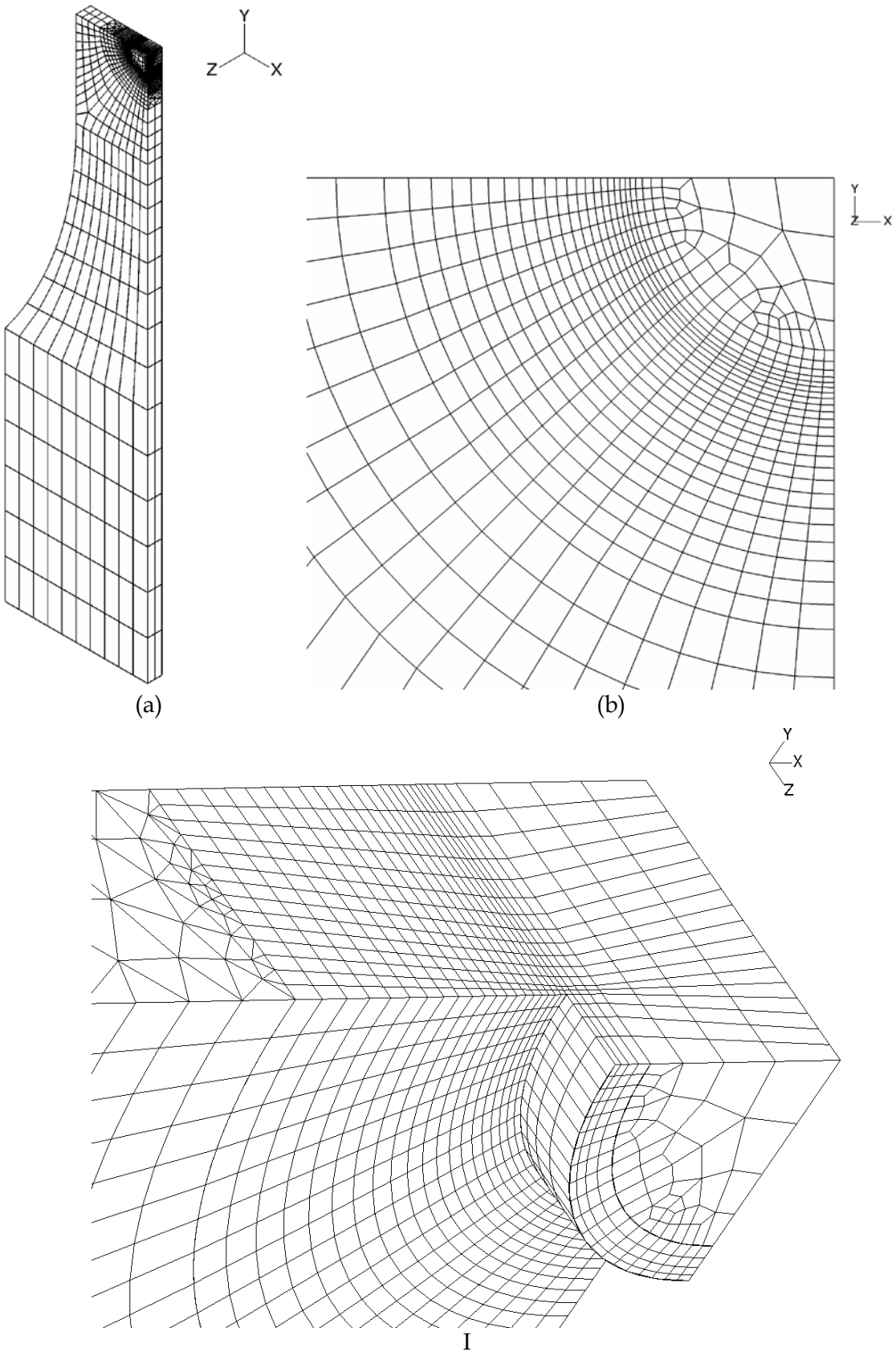
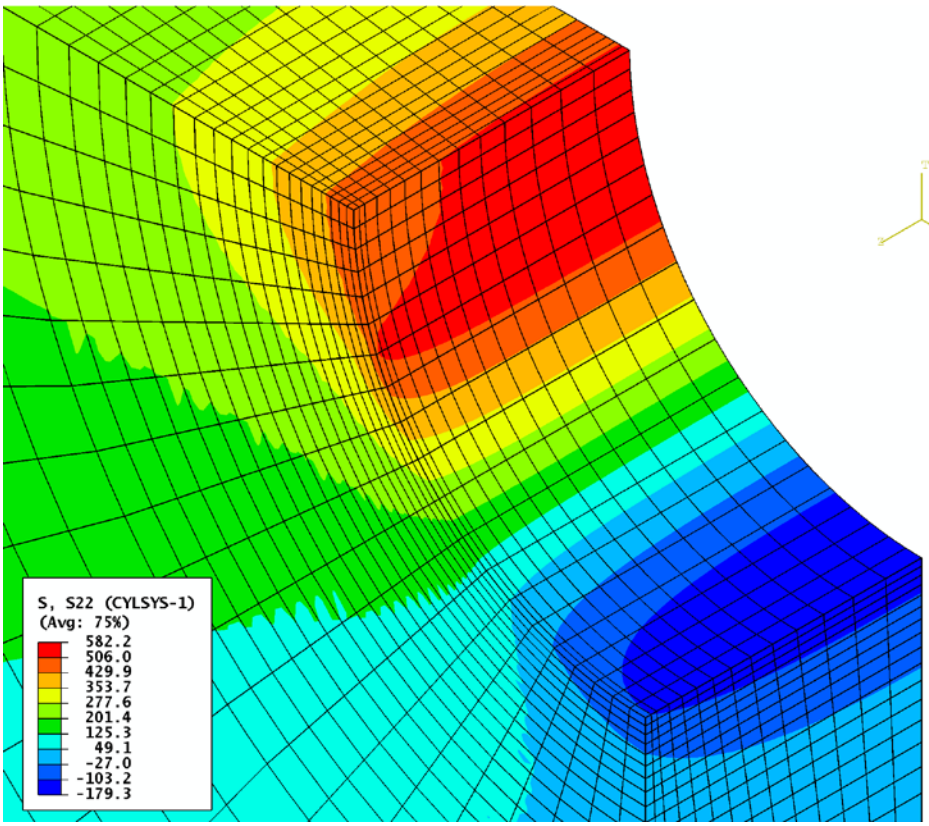
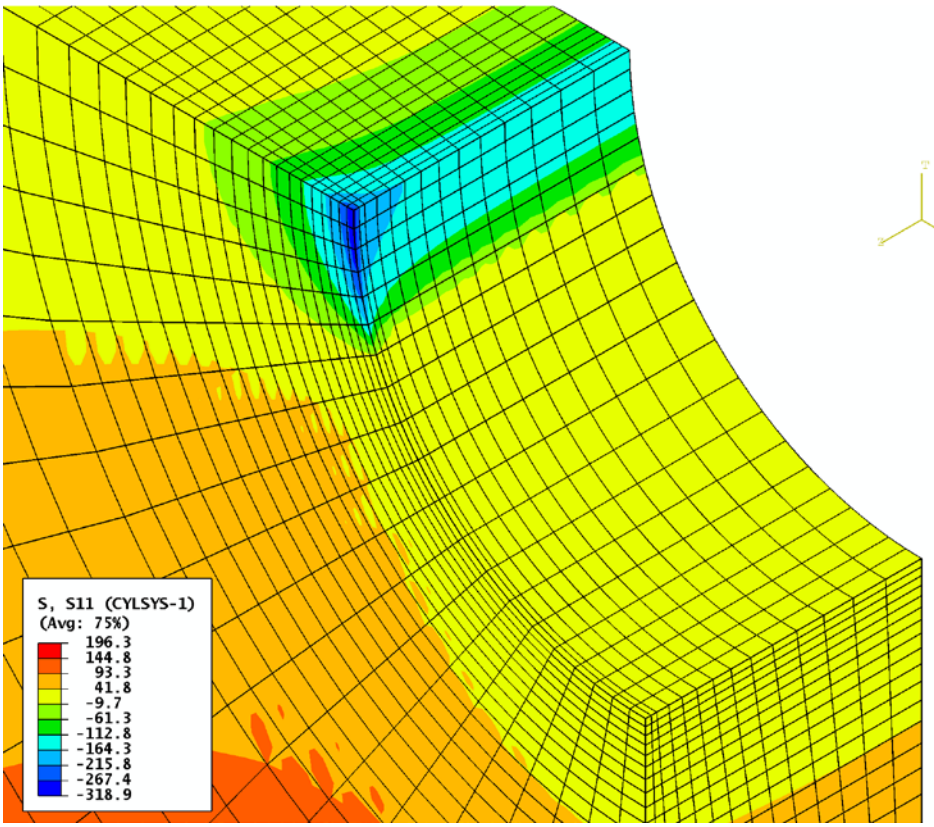


Figure 19: *Abaqus 3D graded-mesh finite element model of the aluminium coupon and the neat-fit titanium pin. (a) Complete ⅛-symmetry model (midplane surface is on the right). (b) Side-on view of plate and pin mesh showing grading around hole–pin boundary. (c) Detail of finite element mesh in vicinity of hole–pin interface.*



(a)



(b)

Figure 20: *Stress contours in the vicinity of the hole for the aluminium coupon with neat-fit titanium pin for the 3D graded finite element mesh. The midplane surface of the coupon is on the right, and the outer (free) surface is on the left. (a) Tangential stress. (b) Radial stress.*

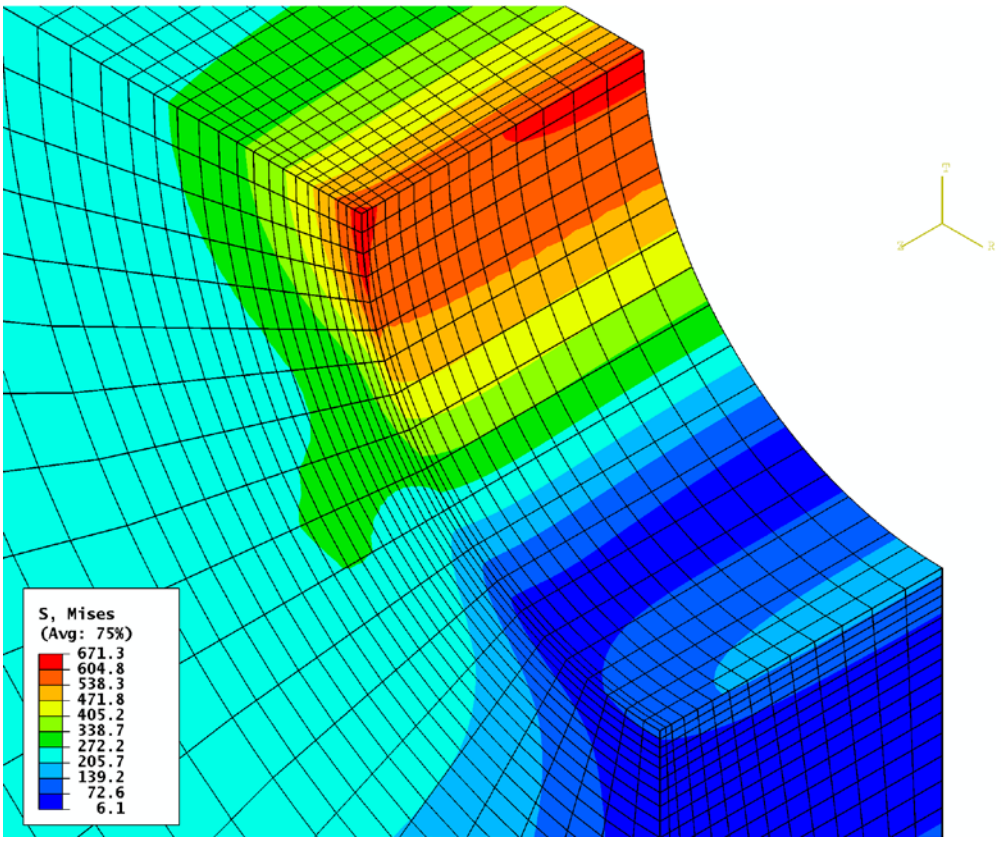
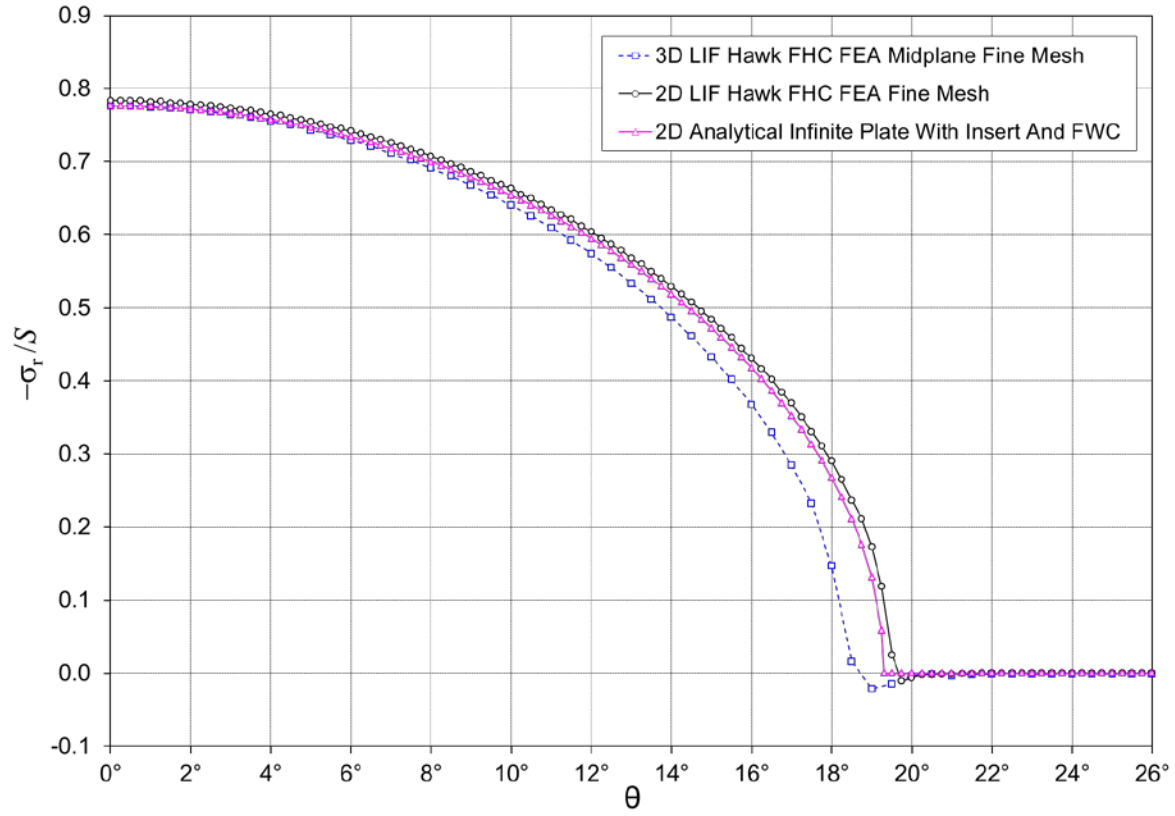
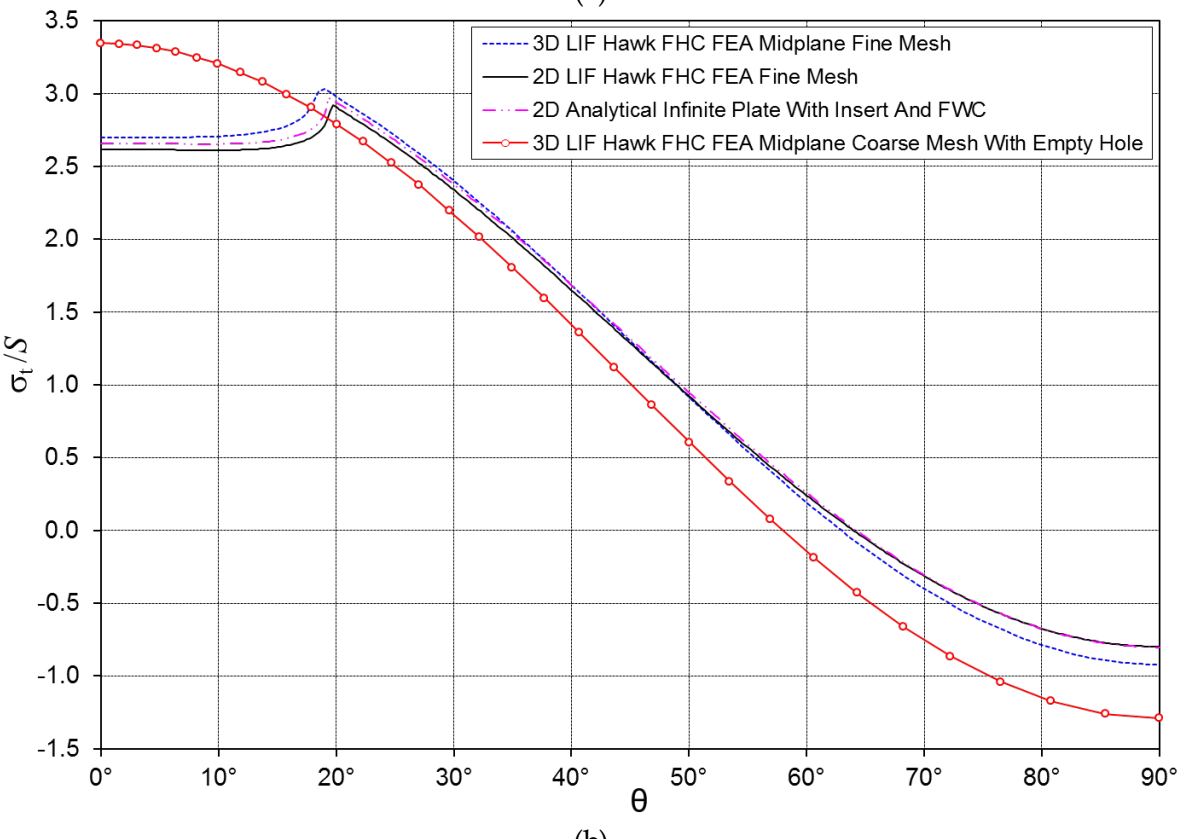


Figure 21: *Von Mises stress contours in the vicinity of the hole for the aluminium coupon with neat-fit titanium pin for the 3D graded finite element mesh, for a 35 kN linear-elastic load level. The midplane surface of the coupon is on the right, and the outer (free) surface is on the left.*

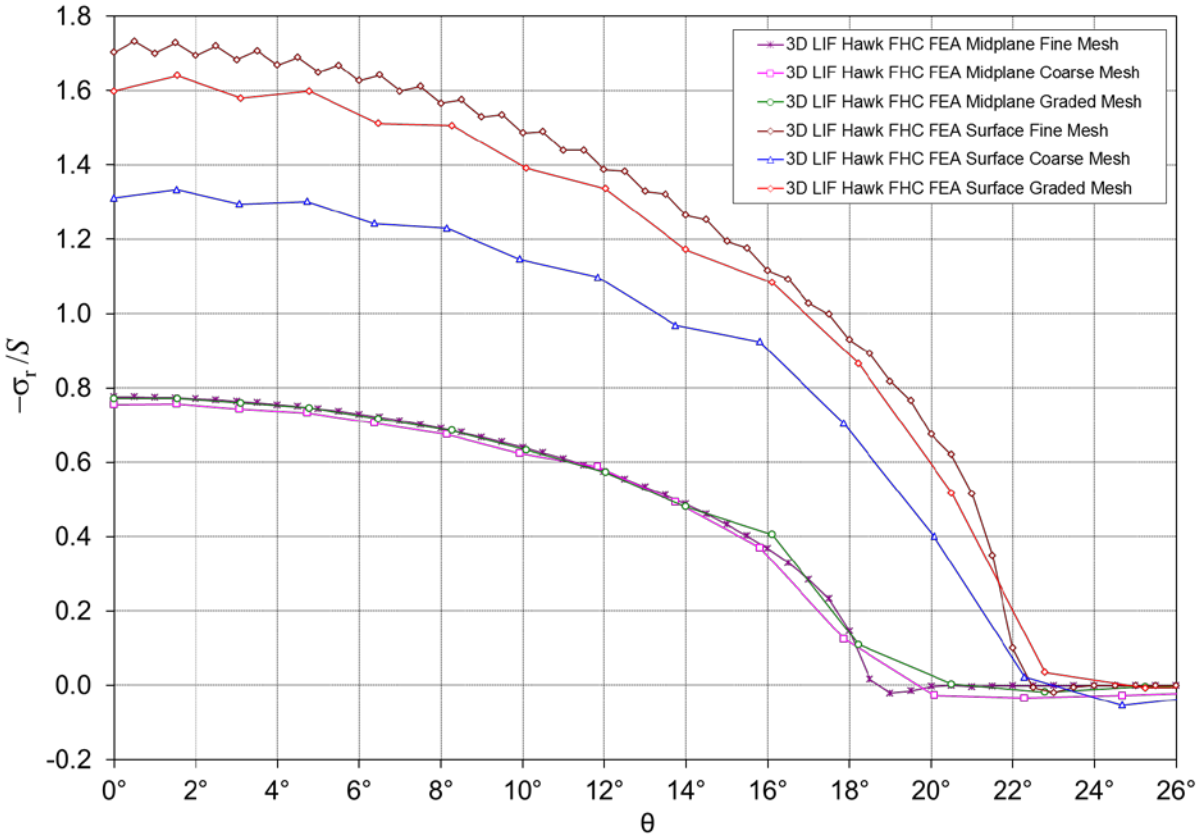


(a)

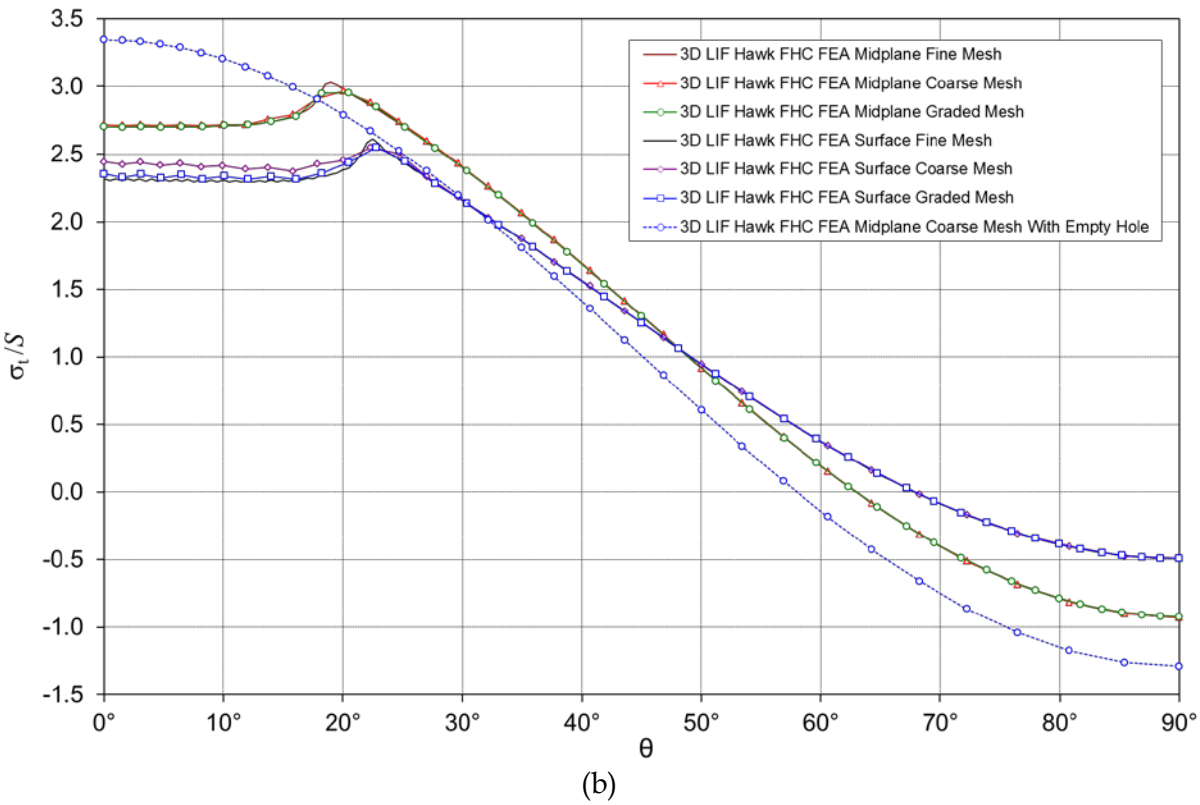


(b)

Figure 22: *Variation of stress around a hole in the uniaxially-loaded aluminium coupon with a neat-fit titanium pin, from Abaqus 2D and 3D FEA midplane (z/t = 0) fine-mesh results and the finite-width-corrected (FWC) analytical infinite-plate solution. (a) Radial stress. (b) Tangential stress.*

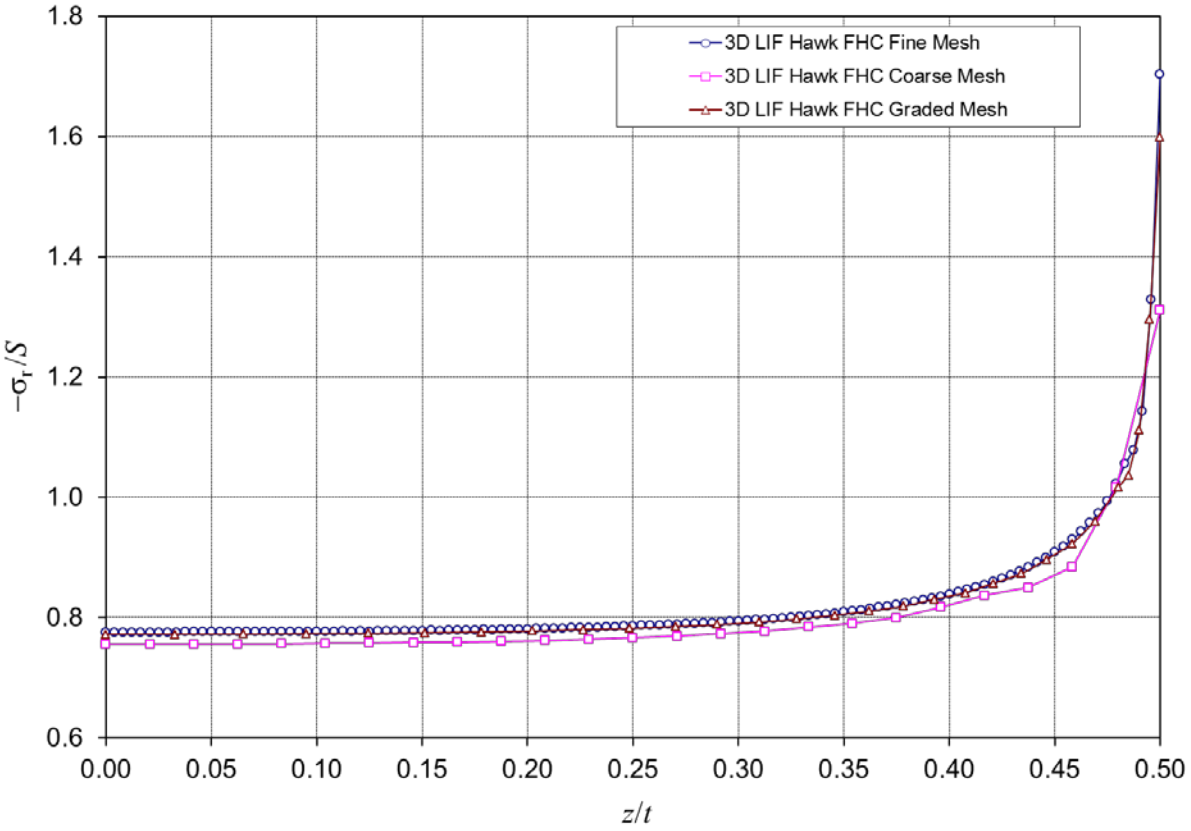


(a)

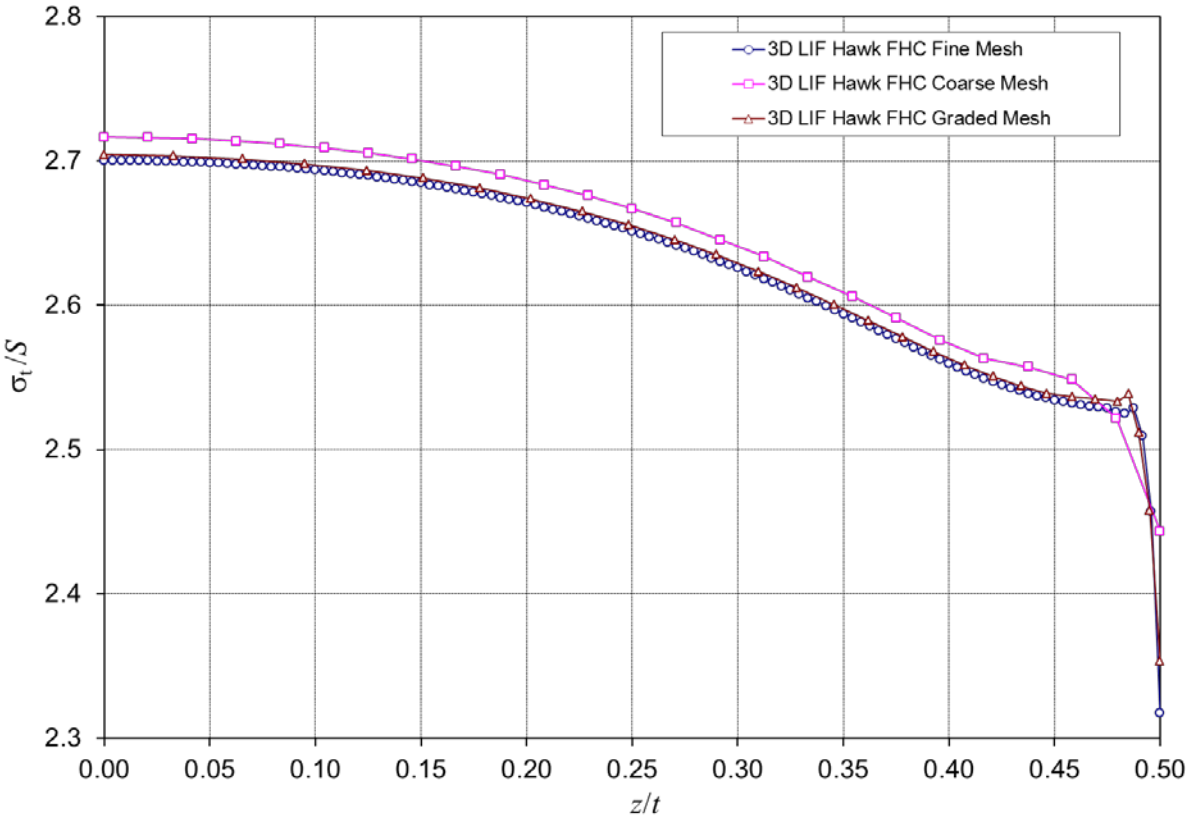


(b)

Figure 23: *Variation of stress around the hole in uniaxially-loaded aluminium coupon with a neat-fit titanium pin, showing Abaqus 3D coarse-mesh, fine-mesh and graded-mesh FEA results at the midplane (z/t = 0.0) and at the free surface (z/t = 0.5) of the coupon. (a) Radial stress. (b) Tangential stress.*



(a)



(b)

Figure 24: *Variation of stress along the bore of the hole in the aluminium coupon with a neat-fit titanium pin, from Abaqus 3D FEA results taken along a line of constant θ = 0°. (a) Radial stress. (b) Tangential stress.*

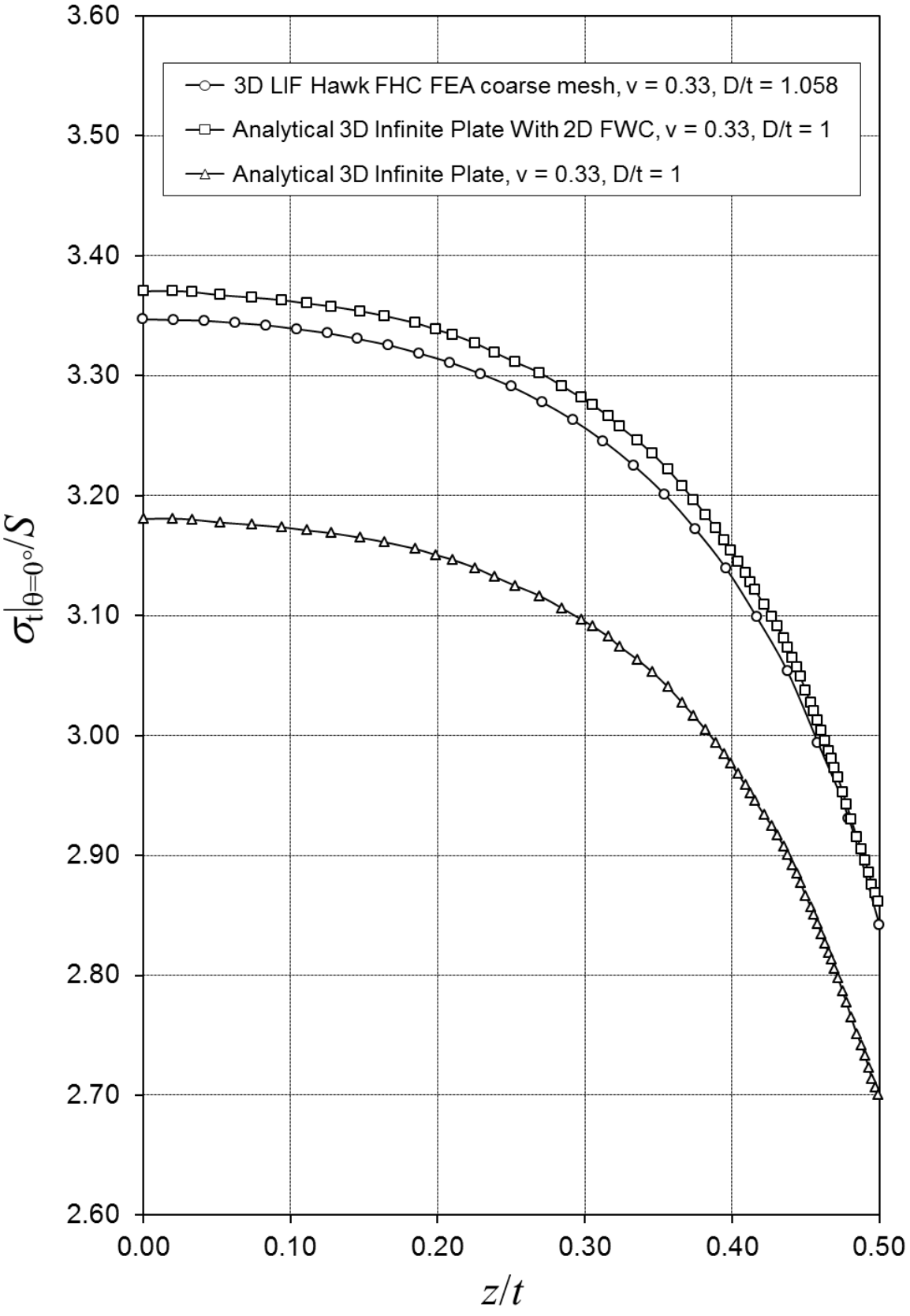


Figure 25: *Stress concentration factor (normalised peak tangential stress, σ_t/S) along the bore of the hole for the aluminium coupon with an empty hole with the coupon loaded in uniaxial tension. Results obtained from Abaqus 3D FEA and a 3D analytical solution [12].*

Appendix A: VBA functions for computing the contact stress distribution around a circular hole containing a circular disk insert

Stippes, Wilson and Krull [2] solved the plane elasticity problem pertaining to the contact between a circular disk and a circular hole in an infinite plate that has a uniaxial stress applied at infinity. The diameter of the disk and the hole are identical in the unstressed state, and the material properties of the disk and the insert are the same. The following two VBA functions are an implementation of their analytical solution, and they can be used to compute the stresses around the hole as a function of angular position. The function `CircularInsertN` is used to compute the radial stress, and the function `CircularInsertStt` is used to compute the tangential stress.

```
'=====================================================================

Function CircularInsertN(ThetaDeg) As Double

Dim A, Pi, EtaDeg, Eta, Theta, N As Double

Application.Volatile

EtaDeg = 19.62506

If ThetaDeg >= EtaDeg Then
  N = 0#
Else
  Pi = 4 * Atn(1)
  Eta = EtaDeg / 180 * Pi
  Theta = ThetaDeg / 180 * Pi
  A = -(2 + 3 * Sin(Eta) ^ 2) / (2 * (2 + Log(Cos(Eta))))
  N = -(-3 / 2 * Cos(Theta) * Sqr(Cos(Theta) ^ 2 - Cos(Eta) ^ 2) + _
  A / 2 * Log((Cos(Theta) + _
  Sqr(Cos(Theta) ^ 2 - Cos(Eta) ^ 2)) / Cos(Eta)))
End If

CircularInsertN = N

End Function

'=====================================================================

Function CircularInsertStt(ThetaDeg) As Double

Dim A, EtaDeg, Eta, Theta, N, Stt As Double

Application.Volatile

EtaDeg = 19.62506
Pi = 4 * Atn(1)
Eta = EtaDeg / 180 * Pi
Theta = ThetaDeg / 180 * Pi
A = -(2 + 3 * Sin(Eta) ^ 2) / (2 * (2 + Log(Cos(Eta))))

If ThetaDeg >= EtaDeg Then
```

```
  N = 0#
Else
  N = -(-3 / 2 * Cos(Theta) * Sqr(Cos(Theta) ^ 2 - Cos(Eta) ^ 2) + _
  A / 2 * Log((Cos(Theta) + _
  Sqr(Cos(Theta) ^ 2 - Cos(Eta) ^ 2)) / Cos(Eta)))
End If

Stt = -N - 2 * A + 2 * Cos(2 * Theta)

CircularInsertStt = Stt

End Function
```

Appendix B: FORTRAN 90 program for two-dimensional contact analysis of a hole containing a circular insert

The following FORTRAN 90 program is an implementation of the solutions to two-dimensional contact problems based on the formulations derived by Stippes, Wilson and Krull [2] and Wilson [3]. The program was created using the Intel Visual FORTRAN Compiler Professional Edition 11.1, and some compiler directives specific to this compiler were used. Apart from the use of the Intel-supplied LAPACK library of numerical subroutines, the program is entirely self-contained. If access to the LAPACK subroutine library is not available, then the compiler directive symbol UseLAPACK can be set equal to 0 (i.e. UseLAPACK == 0) in order to cause the Intel compiler to selectively bypass the small amount of code that relies on calls to some subroutines that are found in LAPACK.

```
!======================================================================

program PlateInsertContactSolutions

! References:
!
! H. B. Wilson, Jr. Approximate Determination of Contact Stresses in
! an Infinite Plate With a Smooth Circular Insert. Proceedings of the
! 2nd Southeastern Conference on Theoretical and Applied Mechanics,
! 5-6 March 1964, Atlanta, Georgia, USA.
!
! M. Stippes, H. B. Wilson, Jr., and F. N. Krull. A contact stress
! problem for a smooth disk in an infinite plate. Proceedings of the
! Fourth US National Congress of Applied Mechanics, Volume 2,
! pages 799-806, 1962.

implicit none

integer   ichoice,lu,iss,p,nOut
character fn*80
real*8    Sx,Sy,E0,E1,nu0,nu1

lu   = 1
fn   = 'PlateInsertContactSolutions.txt'
nOut = 360

open(lu,file=fn,status='unknown')

ichoice = 1

do while (ichoice /= 0)
  write(*,'(a)') &
  '======================================================================'
  write(*,'(a)') &
  'Approximate contact stresses in an infinite plate with smooth insert'
  write(*,'(a)') &
  '======================================================================'
  write(*,'(a)')
  write(*,'(a)') &
  'Results are written to the file "'//fn(1:len_trim(fn))//'".'
  write(*,'(a)') &
  'If a prior version of this file exists, then it is overwritten when'
  write(*,'(a)') &
  'this program starts up.'
  write(*,'(a)')
  write(*,'(a)') &
  '1 Compute solution constants using Wilson''s eta for Cases 1-8'
  write(*,'(a)') &
```

```
'2 Compute full solutions for Wilson''s Cases 1-8'
write(*,'(a)') &
'3 Compute solution for user-defined load and material case'
write(*,'(a)') &
'4 Compute results using data from Stippes, Wilson and Krull'
write(*,'(a)') &
'5 Compute results using data from Wilson''s Cases 1-8'
write(*,'(a)') &
'6 Compute results for unfilled circular hole'
write(*,'(a)') &
'7 Compute convergence study for Wilson''s eta for Cases 1-8'
write(*,'(a,i0,a,i0,a)') &
'8 Input new number of intervals for stress output '// &
'(current n = ',nOut,')'
write(*,'(a)') &
'0 Exit'
write(*,'(a)')
write(*,'(a)',ADVANCE='NO') &
'Input choice: '
read(*,*) ichoice
if (ichoice == 1 .or. ichoice == 2 .or. &
    ichoice == 4 .or. ichoice == 5 .or. &
    ichoice == 6                        ) then
  write(*,'(a)')
  write(*,'(a)') &
  'Computing results and saving to file '//fn(1:len_trim(fn))//'...'
  write(*,'(a)')
end if
select case (ichoice)
  case (1)
    call CheckWilsonCoefficients(lu)
  case (2)
    call ComputeShowWilsonEta(lu,nOut)
  case (3)
    write(*,'(a)')
    write(*,'(a)') &
    'Input the following parameters:'
    write(*,'(a)')
    write(*,'(a)') &
    'iss = stress state (0 = plane stress, 1 = plane strain)'
    write(*,'(a)') &
    'p   = maximum index of constants (A0...Ap, p <= 10)'
    write(*,'(a)') &
    'Sy  = applied stress in y-direction (+ve tension)'
    write(*,'(a)') &
    'Sx  = applied stress in x-direction (+ve tension)'
    write(*,'(a)') &
    'E0  = Young''s Modulus of insert'
    write(*,'(a)') &
    'E1  = Young''s Modulus of plate'
    write(*,'(a)') &
    'nu0 = Poisson''s Ratio of insert'
    write(*,'(a)') &
    'nu0 = Poisson''s Ratio of plate'
    write(*,'(a)')
    write(*,'(a)', ADVANCE='NO') &
    'Input iss, p, Sy, Sx, E0, E1, nu0, nu1: '
    read(*,*) iss, p, Sy, Sx, E0, E1, nu0, nu1
    write(*,'(a)')
    write(*,'(a)') &
    'Computing results and saving to file '//fn(1:len_trim(fn))//'...'
    write(*,'(a)')
    p = max(min(p,10),0)
    call ComputeShowWilsonUD(lu,nOut,iss,p,Sx,Sy,E0,E1,nu0,nu1)
  case (4)
    call ComputeShowSWK(lu,nOut)
  case (5)
    call ComputeWilsonCases1to8(lu,nOut)
```

```
      case (6)
        call ComputeShowEmptyHole(lu,nOut)
      case (7)
        call WilsonEtaConvergence(lu)
      case (8)
        write(*,'(a)')
        write(*,'(a,i0,a)',ADVANCE='NO') &
        'Number of intervals for computing stresses (current nOut = ', &
        nOut,') = '
        read(*,*) nOut
        nOut = max(nOut,20)
    end select
    write(*,'(a)')
  end do

  close(lu)

  stop
  end

  !=====================================================================

  subroutine ComputeShowEmptyHole(lu,nOut)

  implicit none

  integer lu,nOut

  integer i
  real*8  thetadeg

  real*8  FuncSttHole

  write(lu,*)
  write(lu,'(a,i1)') 'NORMALISED HOOP STRESS FOR CIRCULAR HOLE IN A PLATE'
  write(lu,'(a)'   ) '================================================='
  write(lu,*)
  write(lu,'(2a15)') 'Theta_deg','Stt'
  do I = 0,nOut
    thetadeg = i*90.0d0/nOut
    write(lu,'(f15.6,f15.8)') thetadeg,FuncSttHole(thetadeg)
  end do

  return
  end subroutine

  !=====================================================================

  real*8 function FuncSttHole(thetadeg)

  implicit none

  real*8 thetadeg

  FuncSttHole = 1.0d0 - 2.0d0*cosd(2.0d0*(thetadeg+90.0d0))

  return
  end function

  !=====================================================================

  real*8 function SWKFuncEqn45(etadeg)

  implicit none

  real*8  etadeg

  real*8  EIK,EIE,k,LHS,RHS,m
```

```
k = cosd(etadeg)
m = k**2

call CompleteEllipticIntegrals12(m,EIK,EIE)

LHS = 4.0d0*sind(etadeg)**2*(2.0d0+log(cosd(etadeg)))*EIK

RHS = (2.0d0+sind(etadeg)**2 + &
       2.0d0*(1.0d0+sind(etadeg)**2)*log(cosd(etadeg)))*EIE

SWKFuncEqn45 = LHS - RHS

return
end function

!=======================================================================

subroutine ComputeShowSWK(lu,nOut)

implicit none

integer lu,nOut
real*8  Sy

integer I,iflag
real*8  etadeg,thetadeg,thetanext
real*8  b,c,r,re,ae,resb,resc

real*8  SWKFuncNtt, SWKFuncStt, SWKFuncEqn45

r  = 1.0d0
re = 1.0d-07
ae = 0.0d0

! The residual function typically has multiple zeros in the range
! [0°, 90°]. It is necessary to look for the smallest value of eta
! within this range. Here we start scanning for a change in sign
! from eta = 1.0 using 1-degree steps. The lowest value of eta
! will then lie in the range [b, c].

b    = 1.0d0
resb = SWKFuncEqn45(b)
do c = 2.0d0, 90.0d0, 1.0d0
  resc = SWKFuncEqn45(c)
  if (nint(sign(1.0d0,resb)) == nint(sign(1.0d0,resc))) then
    b    = c
    resb = resc
  else
    exit
  end if
end do

call DFZERO(SWKFuncEqn45, b, c, r, re, ae, iflag)

etadeg = b

Sy = 1.0d0

write(lu,*)
write(lu,'(a,i1)') &
'ANALYTICAL CONTACT STRESSES FOR PLATE WITH SMOOTH CIRCULAR INSERT'
write(lu,'(a)'   ) &
'================================================================'
write(lu,*)
write(lu,'(a)'   ) &
'Analytical solution by Stippes, Wilson and Krull (1962) for the'
write(lu,'(a)'   ) &
```

```
    'case of a uniaxially loaded plate with an insert made from the'
    write(lu,'(a)'   ) &
    'same material as that of the plate.'
    write(lu,*)
    write(lu,'(a,es14.06)') 'Sy  = ',1.0
    write(lu,'(a,es14.06)') 'Sx  = ',0.0
    write(lu,'(a,es14.06)') 'E0  = E1'
    write(lu,'(a, f14.10)') 'nu0 = nu1'
    write(lu,*)
    write(lu,'(a,f14.10)') 'Eta         = ',etadeg
    write(lu,'(a,f14.10)') 'Residual    = ',SWKFuncEqn45(etadeg)
    write(lu,'(a,i14   )') 'DFZERO flag = ',iflag
    write(lu,*)
    write(lu,'(a,f14.10)') 'Ntt(  0)    = ',SWKFuncNtt( 0.0d0,etadeg,Sy)
    write(lu,'(a,f14.10)') 'Stt(  0)    = ',SWKFuncStt( 0.0d0,etadeg,Sy)
    write(lu,'(a,f14.10)') 'Stt(Eta)    = ',SWKFuncStt(etadeg,etadeg,Sy)
    write(lu,'(a,f14.10)') 'Stt( 90)    = ',SWKFuncStt(90.0d0,etadeg,Sy)

    if (nOut > 0) then
      write(lu,*)
      write(lu,'(3a15)') 'Theta_deg','Stt','Ntt'
      do I = 0,nOut
        thetadeg = i*90.0d0/nOut
        write(lu,'(f15.6,2f15.8)') thetadeg, &
          SWKFuncStt(thetadeg,etadeg,Sy), SWKFuncNtt(thetadeg,etadeg,Sy)
        thetanext = (i+1)*90.0d0/nOut
        if (etadeg > thetadeg .and. etadeg < thetanext) then
          write(lu,'(f15.6,2f15.8)') etadeg, &
            SWKFuncStt(etadeg,etadeg,Sy), SWKFuncNtt(etadeg,etadeg,Sy)
        end if
      end do
    end if

    return
    end subroutine

    !=======================================================================

    real*8 function SWKFuncNtt(ThetaDeg,EtaDeg,Sy)

    implicit none

    real*8  ThetaDeg,EtaDeg,Sy

    real*8  Ntt,sc2tmc2e,A

    if (ThetaDeg < EtaDeg) then
      A = -Sy*(2.0d0+3.0d0*sind(EtaDeg)**2)/(2.0d0*(2.0d0+log(cosd(EtaDeg))))
      sc2tmc2e = sqrt(cosd(ThetaDeg)**2-cosd(EtaDeg)**2)
      Ntt = -3.0d0/2.0d0*Sy*cosd(ThetaDeg)*sc2tmc2e + &
            A/2.0d0*log((cosd(ThetaDeg)+sc2tmc2e)/cosd(EtaDeg))
    else
      Ntt = 0.0d0
    end if

    SWKFuncNtt = Ntt

    return
    end function

    !=======================================================================

    real*8 function SWKFuncB(EtaDeg,Sy)

    implicit none

    real*8  EtaDeg,Sy
```

```
real*8  A,B

A = -Sy*(2.0d0+3.0d0*sind(EtaDeg)**2)/(2.0d0*(2.0d0+log(cosd(EtaDeg))))

B = -1.0d0/2.0d0*(A+Sy/2.0d0)

SWKFuncB = B

return
end function

!=====================================================================

real*8 function SWKFuncStt(ThetaDeg, EtaDeg, Sy)

implicit none

real*8  ThetaDeg, EtaDeg, Sy

real*8  Ntt, Stt, B

real*8 SWKFuncNtt, SWKFuncB

Ntt = SWKFuncNtt(ThetaDeg, EtaDeg, Sy)
B   = SWKFuncB(EtaDeg,Sy)
Stt = Ntt + 4.0d0*B + Sy + 2.0d0*Sy*cosd(2.0d0*ThetaDeg)

SWKFuncStt = Stt

return
end function

!=====================================================================

subroutine ComputeShowWilsonUD(lu,nOut,iss,p,Sx,Sy,E0,E1,nu0,nu1)

implicit none

integer   lu,iss,p,nOut
real*8    Sx,Sy,E0,E1,nu0,nu1

integer   I,iflag
real*8    A(0:p),etadeg,etarad,pi,Stt,Ntt
real*8    b,c,r,re,ae,resb,resc,thetadeg,rcond,thetanext
character istr*2

real*8    FuncNtt
real*8    FuncStt
real*8    FuncEqn35

external  FuncEqn35

real*8  cmn_Sx,cmn_Sy,cmn_nu0,cmn_nu1,cmn_E0,cmn_E1
integer cmn_p,cmn_iss

common /cmn_FuncEqn35/ cmn_Sx,cmn_Sy,cmn_nu0,cmn_nu1, &
                       cmn_E0,cmn_E1,cmn_p,cmn_iss

cmn_Sx  = Sx
cmn_Sy  = Sy
cmn_nu0 = nu0
cmn_nu1 = nu1
cmn_E0  = E0
cmn_E1  = E1
cmn_p   = p
cmn_iss = iss

pi = 4.0d0*atan(1.0d0)
```

```
r  =  1.0d0
re =  1.0d-08
ae =  0.0d0

! The residual function typically has multiple zeros in the range
! [0°, 90°]. It is necessary to look for the smallest value of eta
! within this range. Here we start scanning for a change in sign
! from eta = 1.0 using 1-degree steps. The lowest value of eta
! will then lie in the range [b, c].

b    = 1.0d0
resb = FuncEqn35(b)
do c = 2.0d0, 90.0d0, 1.0d0
  resc = FuncEqn35(c)
  if (nint(sign(1.0d0,resb)) == nint(sign(1.0d0,resc))) then
    b    = c
    resb = resc
  else
    exit
  end if
end do

call DFZERO(FuncEqn35, b, c, r, re, ae, iflag)

etadeg = b
etarad = etadeg/180.0d0*pi

call GetA0toAp(p,iss,etarad,Sx,Sy,nu0,nu1,E0,E1,A,rcond)

write(lu,*)
write(lu,'(a,i1)') &
'APPROXIMATE CONTACT STRESSES FOR PLATE WITH SMOOTH CIRCULAR INSERT'
write(lu,'(a)'   ) &
'================================================================='
write(lu,*)
if (iss == 0) then
  write(lu,'(a,i14,a)') 'iss          = ',iss,' (plane stress)'
else
  write(lu,'(a,i14,a)') 'iss          = ',iss,' (plane strain)'
endif
write(lu,'(a,es14.06)') 'Sy           = ',Sy
write(lu,'(a,es14.06)') 'Sx           = ',Sx
write(lu,'(a,es14.06)') 'E0           = ',E0
write(lu,'(a,es14.06)') 'E1           = ',E1
write(lu,'(a, f14.10)') 'nu0          = ',nu0
write(lu,'(a, f14.10)') 'nu1          = ',nu1
write(lu,*)
write(lu,'(a, f14.10)') 'Eta          = ',etadeg
write(lu,'(a, f14.10)') 'Residual     = ',FuncEqn35(etadeg)
write(lu,'(a, i14   )') 'DFZERO flag  = ',iflag
write(lu,'(a,es14.03)') 'Matrix rcond = ',rcond
write(lu,*)
do I = 0, p
  write(istr,'(i2)') i
  if (istr(1:1) == ' ') istr = istr(2:2)
  write(lu,'(a,a,a,f14.10)') 'A',istr,'         = ',A(i)
end do
write(lu,*)
write(lu,'(a,f14.10)') 'Ntt(  0)     = ',FuncNtt( 0.0d0,etadeg,p,A)
write(lu,'(a,f14.10)') 'Stt(  0)     = ',FuncStt( 0.0d0,etadeg,Sx,Sy,p,A)
write(lu,'(a,f14.10)') 'Stt(Eta)     = ',FuncStt(EtaDeg,etadeg,Sx,Sy,p,A)
write(lu,'(a,f14.10)') 'Stt( 90)     = ',FuncStt(90.0d0,etadeg,Sx,Sy,p,A)

if (nOut > 0) then
  write(lu,*)
  write(lu,'(3a15)') 'Theta_deg','Stt','Ntt'
  do I = 0,nOut
```

```
    thetadeg = i*90.0d0/nOut
    Stt = FuncStt(thetadeg,etadeg,Sx,Sy,p,A)
    Ntt = FuncNtt(thetadeg,etadeg,p,A)
    write(lu,'(f15.6,2f15.8)') thetadeg, Stt, Ntt
    thetanext = (i+1)*90.0d0/nOut
    if (etadeg > thetadeg .and. etadeg < thetanext) then
      Stt = FuncStt(etadeg,etadeg,Sx,Sy,p,A)
      Ntt = FuncNtt(etadeg,etadeg,p,A)
      write(lu,'(f15.6,2f15.8)') etadeg, Stt, Ntt
    end if
  end do
end if

return
end subroutine

!=====================================================================

subroutine WilsonEtaConvergence(lu)

implicit none

integer   lu

integer   pmax

parameter (pmax = 10)

integer   iss,I,iflag,caseno,p,pwilson
real*8    Sx,Sy,nu0,nu1,E0,E1,A(0:pmax),etadeg,etarad,etawilson,pi
real*8    b,c,r,re,ae,resb,resc,rcond

real*8    FuncEqn35

external  FuncEqn35

common /cmn_FuncEqn35/ Sx,Sy,nu0,nu1,E0,E1,p,iss

pi = 4.0d0*atan(1.0d0)

write(*,*)

! Compute coefficients for Wilson's Cases 1 to 8.

do caseno = 1, 8

  write(*,'(a,i1.0,a,/)') &
    'Computing eta convergence study for Wilson''s Case ',caseno,'...'

  write(lu,*)
  write(lu,'(a,i1)') 'ETA CONVERGENCE FOR WILSON''S CASE ',caseno
  write(lu,'(a)'   ) '================================='
  write(lu,*)
  write(lu,'(a4,a17,a11,a10,<pmax+1>(a12,i2.2))') &
    'p','EtaDeg','Residual','rcond',('A',I,i=0,pmax)

  do p = 0,pmax

    write(*,'(a,i4)') '  Doing p =',p

    call GetWilsonData(caseno,pwilson,iss,etawilson,Sx,Sy,nu0,nu1,E0,E1)

    r  =  1.0d0
    re =  1.0d-08
    ae =  0.0d0

    ! The residual function typically has multiple zeros in the range
    ! [0°, 90°]. It is necessary to look for the smallest value of eta
```

```
      ! within this range. Here we start scanning for a change in sign
      ! from eta = 1.0 using 1-degree steps. The lowest value of eta
      ! will then lie in the range [b, c].

      b    = 1.0d0
      resb = FuncEqn35(b)
      do c = 2.0d0, 90.0d0, 1.0d0
        resc = FuncEqn35(c)
        if (nint(sign(1.0d0,resb)) == nint(sign(1.0d0,resc))) then
          b    = c
          resb = resc
        else
          exit
        end if
      end do

      call DFZERO(FuncEqn35, b, c, r, re, ae, iflag)

      etadeg = b
      etarad = etadeg/180.0d0*pi

      call GetA0toAp(p,iss,etarad,Sx,Sy,nu0,nu1,E0,E1,A,rcond)

      write(lu,'(i4,f17.12,es11.2,es10.2,<pmax+1>es14.6)') &
        p,etadeg,FuncEqn35(etadeg),rcond,(A(i),i=0,p)

    end do

    write(*,*)

  end do

  return
  end subroutine

  !=====================================================================

  subroutine ComputeShowWilsonEta(lu,nOut)

  implicit none

  integer   lu,nOut

  integer   pmax

  parameter (pmax = 10)

  integer   iss,I,iflag,caseno,p
  real*8    Sx,Sy,nu0,nu1,E0,E1,A(0:pmax),etadeg,etarad,etawilson,pi
  real*8    b,c,r,re,ae,resb,resc,thetadeg,rcond,thetanext,Stt,Ntt
  character istr*2

  real*8    FuncNtt
  real*8    FuncStt
  real*8    FuncEqn35

  external  FuncEqn35

  common /cmn_FuncEqn35/ Sx,Sy,nu0,nu1,E0,E1,p,iss

  pi = 4.0d0*atan(1.0d0)

  ! Compute coefficients for Wilson's Cases 1 to 8.

  do caseno = 1, 8

    write(*,'(a,i1.0,a)') 'Computing Wilson''s Case ',caseno,'...'
```

```
call GetWilsonData(caseno,p,iss,etawilson,Sx,Sy,nu0,nu1,E0,E1)

r  =  1.0d0
re =  1.0d-08
ae =  0.0d0

! The residual function typically has multiple zeros in the range
! [0°, 90°]. It is necessary to look for the smallest value of eta
! within this range. Here we start scanning for a change in sign
! from eta = 1.0 using 1-degree steps. The lowest value of eta
! will then lie in the range [b, c].

b    = 1.0d0
resb = FuncEqn35(b)
do c = 2.0d0, 90.0d0, 1.0d0
  resc = FuncEqn35(c)
  if (nint(sign(1.0d0,resb)) == nint(sign(1.0d0,resc))) then
    b    = c
    resb = resc
  else
    exit
  end if
end do

call DFZERO(FuncEqn35, b, c, r, re, ae, iflag)

etadeg = b
etarad = etadeg/180.0d0*pi

call GetA0toAp(p,iss,etarad,Sx,Sy,nu0,nu1,E0,E1,A,rcond)

write(lu,*)
write(lu,'(a,i1)') 'COMPUTED RESULTS FOR WILSON''S CASE ',caseno
write(lu,'(a)'   ) '==================================='
write(lu,*)
if (iss == 0) then
  write(lu,'(a,i14,a)') 'iss          = ',iss,' (plane stress)'
else
  write(lu,'(a,i14,a)') 'iss          = ',iss,' (plane strain)'
endif
write(lu,'(a,es14.06)') 'Sy           = ',Sy
write(lu,'(a,es14.06)') 'Sx           = ',Sx
write(lu,'(a,es14.06)') 'E0           = ',E0
write(lu,'(a,es14.06)') 'E1           = ',E1
write(lu,'(a, f14.10)') 'nu0          = ',nu0
write(lu,'(a, f14.10)') 'nu1          = ',nu1
write(lu,*)
write(lu,'(a, f14.10)') 'Eta Present  = ',etadeg
write(lu,'(a, f14.10)') 'Eta Wilson   = ',etawilson
write(lu,'(a, f14.10)') 'Residual     = ',FuncEqn35(etadeg)
write(lu,'(a, i14   )') 'DFZERO flag  = ',iflag
write(lu,'(a,es14.03)') 'Matrix rcond = ',rcond
write(lu,*)
do I = 0, p
  write(istr,'(i2)') i
  if (istr(1:1) == ' ') istr = istr(2:2)
  write(lu,'(a,a,a,f14.10)') 'A',istr,'          = ',A(i)
end do
write(lu,*)
write(lu,'(a,f14.10)') 'Ntt(  0)     = ', &
  FuncNtt( 0.0d0,etadeg,p,A)
write(lu,'(a,f14.10)') 'Stt(  0)     = ', &
  FuncStt( 0.0d0,etadeg,Sx,Sy,p,A)
write(lu,'(a,f14.10)') 'Stt(Eta)     = ', &
  FuncStt(EtaDeg,etadeg,Sx,Sy,p,A)
write(lu,'(a,f14.10)') 'Stt( 90)     = ', &
  FuncStt(90.0d0,etadeg,Sx,Sy,p,A)
```

```
      if (nOut > 0) then
        write(lu,*)
        write(lu,'(3a15)') 'Theta_deg','Stt','Ntt'
        do I = 0,nOut
          thetadeg = i*90.0d0/nOut
          Stt = FuncStt(thetadeg,etadeg,Sx,Sy,p,A)
          Ntt = FuncNtt(thetadeg,etadeg,p,A)
          write(lu,'(f15.6,2f15.8)') thetadeg, Stt, Ntt
          thetanext = (i+1)*90.0d0/nOut
          if (etadeg > thetadeg .and. etadeg < thetanext) then
            Stt = FuncStt(etadeg,etadeg,Sx,Sy,p,A)
            Ntt = FuncNtt(etadeg,etadeg,p,A)
            write(lu,'(f15.6,2f15.8)') etadeg, Stt, Ntt
          end if
        end do
      end if

    end do

    return
    end subroutine

    !=====================================================================

    real*8 function FuncEqn35(x)

    implicit none

    real*8 x

    integer p,iss,n
    real*8  Sx,Sy,nu0,nu1,E0,E1
    real*8  A(0:p),lhs,rhs,eta,k1,k2,k3,pi,rcond

    real*8   Eneta,Fneta,Dstarnp1
    external Eneta,Fneta,Dstarnp1

    common /cmn_FuncEqn35/ Sx,Sy,nu0,nu1,E0,E1,p,iss

    pi = 4.0d0*atan(1.0d0)

    eta = x/180.0d0*pi

    call GetA0toAp(p,iss,eta,Sx,Sy,nu0,nu1,E0,E1,A,rcond)

    call Computek1k2k3(k1,k2,k3,iss,nu0,nu1,E0,E1)

    lhs = 0.0d0
    do n = 0, p
      lhs = lhs + &
            (k1*Eneta(n,eta) + k2*Fneta(n,eta) - k3*Dstarnp1(n,eta))*A(n)
    end do

    rhs = Sy - 3.0d0*Sx

    FuncEqn35 = lhs - rhs

    return
    end function

    !=====================================================================

    real*8 function WilsonFuncEqn35(x,iss,Sx,Sy,nu0,nu1,E0,E1,p,A)

    implicit none

    real*8   x
```

```
integer  p,iss,n
real*8   Sx,Sy,nu0,nu1,E0,E1
real*8   A(0:p),lhs,rhs,eta,k1,k2,k3,pi

real*8   Eneta,Fneta,Dstarnp1
external Eneta,Fneta,Dstarnp1

pi = 4.0d0*atan(1.0d0)

eta = x/180.0d0*pi

call Computek1k2k3(k1,k2,k3,iss,nu0,nu1,E0,E1)

rhs = Sy - 3.0d0*Sx

lhs = 0.0d0
do n = 0, p
  lhs = lhs + &
        (k1*Eneta(n,eta) + k2*Fneta(n,eta) - k3*Dstarnp1(n,eta))*A(n)
end do

WilsonFuncEqn35 = lhs - rhs

return
end function

!=====================================================================

subroutine Computek1k2k3(k1,k2,k3,iss,nu0,nu1,E0,E1)

! iss  = 0 for plane stress
! iss /= 0 for plane strain
!
! nu0 = Poisson's Ratio for insert
! nu1 = Poisson's Ratio for plate
! E0  = Young's Modulus for insert
! E1  = Young's Modulus for plate

implicit none

real*8  k1,k2,k3,nu0,nu1,E0,E1
integer iss

real*8  N0,N1,mu0,mu1

if (iss == 0) then
  N0 = (3.0d0-nu0)/(1.0d0+nu0)
  N1 = (3.0d0-nu1)/(1.0d0+nu1)
else
  N0 = 3.0d0-4.0d0*nu0
  N1 = 3.0d0-4.0d0*nu1
end if

mu0 = E0/(2.0d0*(1+nu0))
mu1 = E1/(2.0d0*(1+nu1))

k1 = (2.0d0*(1.0d0-N0)*mu1-2.0d0*(1.0d0-N1)*mu0)/((1.0d0+N1)*mu0)
k2 = (2.0d0*(1.0d0+N0)*mu1+2.0d0*(1.0d0+N1)*mu0)/((1.0d0+N1)*mu0)
k3 = (4.0d0*(1.0d0+N0)*mu1)/((1.0d0+N1)*mu0)

return
end subroutine

!=====================================================================

subroutine ComputeShowWilson(lu,caseno,p,iss,etadeg,Sx,Sy,nu0,nu1,E0,E1)

implicit none
```

```
integer   lu,caseno,p,iss
real*8    etadeg,Sx,Sy,nu0,nu1,E0,E1

real*8    A(0:4),eta,pi,rcond,zero,ninety
integer   i
character istr*2

real*8    FuncNtt
real*8    FuncStt

pi     = 4.0d0*atan(1.0d0)
zero   = 0.0d0
ninety = 90.0d0

eta = etadeg/180.0d0*pi

call GetA0toAp(p,iss,eta,Sx,Sy,nu0,nu1,E0,E1,A,rcond)

write(lu,*)
write(lu,'(a,i1)') &
  'COMPUTED CONSTANTS USING WILSON''S ETA FOR CASE ',caseno
write(lu,'(a)'   ) &
  '================================================'
write(lu,*)
if (iss == 0) then
  write(lu,'(a,i14,a)') 'iss = ',iss,' (plane stress)'
else
  write(lu,'(a,i14,a)') 'iss = ',iss,' (plane strain)'
endif
write(lu,'(a,es14.06)') 'Sy  = ',Sy
write(lu,'(a,es14.06)') 'Sx  = ',Sx
write(lu,'(a,es14.06)') 'E0  = ',E0
write(lu,'(a,es14.06)') 'E1  = ',E1
write(lu,'(a, f14.10)') 'nu0 = ',nu0
write(lu,'(a, f14.10)') 'nu1 = ',nu1
write(lu,*)
write(lu,'(a,f10.6,a)') 'Eta          = ',etadeg,' degrees'
write(lu,'(a,es10.03)') 'Matrix rcond = ',rcond
write(lu,*)
do I = 0, p
  write(istr,'(i2)') i
  if (istr(1:1) == ' ') istr = istr(2:2)
  write(lu,'(a,a,a,f10.6)') 'A',istr,'      = ',A(i)
end do
write(lu,*)
write(lu,'(a,f10.6)') 'Ntt(  0) = ',FuncNtt(  zero,EtaDeg,p,A)
write(lu,'(a,f10.6)') 'Stt(  0) = ',FuncStt(  zero,EtaDeg,Sx,Sy,p,A)
write(lu,'(a,f10.6)') 'Stt(Eta) = ',FuncStt(EtaDeg,EtaDeg,Sx,Sy,p,A)
write(lu,'(a,f10.6)') 'Stt( 90) = ',FuncStt(ninety,EtaDeg,Sx,Sy,p,A)

return
end

!=====================================================================

subroutine GetWilsonData(caseno,p,iss,etadeg,Sx,Sy,nu0,nu1,E0,E1)

integer caseno,p,iss
real*8  etadeg,Sx,Sy,nu0,nu1,E0,E1

select case (caseno)
  case (1)
    iss     =  1
    p       =  3
    etadeg  =  17.285975d0
    Sx      =  0.0d0
    Sy      = +1.0d0
```

```
  nu0     =  0.30d0
  nu1     =  0.50d0
  E0      =  70.0d9
  E1      =  E0*1.0d-09
case (2)
  iss     =  1
  p       =  3
  etadeg  =  19.625247d0
  Sx      =  0.0d0
  Sy      = +1.0d0
  nu0     =  0.30d0
  nu1     =  nu0
  E0      =  70.0d9
  E1      =  E0
case (3)
  iss     =  1
  p       =  4
  etadeg  =  18.338752d0
  Sx      =  0.0d0
  Sy      = +1.0d0
  nu0     =  0.30d0
  nu1     =  0.30d0
  E0      =  70.0d9
  E1      =  E0*1.0d-09
case (4)
  iss     =  1
  p       =  3
  etadeg  =  20.580669d0
  Sx      =  0.0d0
  Sy      = +1.0d0
  nu0     =  0.33d0
  nu1     =  0.29d0
  E0      =  70.0d9
  E1      =  E0*3.0d0
case (5)
  iss     =  1
  p       =  4
  etadeg  =  40.472075d0
  Sx      = -1.0d0
  Sy      =  0.0d0
  nu0     =  0.30d0
  nu1     =  0.50d0
  E0      =  70.0d9
  E1      =  E0*1.0d-09
case (6)
  iss     =  1
  p       =  4
  etadeg  =  56.973112d0
  Sx      = -1.0d0
  Sy      =  0.0d0
  nu0     =  0.30d0
  nu1     =  nu0
  E0      =  70.0d9
  E1      =  E0
case (7)
  iss     =  1
  p       =  4
  etadeg  =  49.813774d0
  Sx      = -1.0d0
  Sy      =  0.0d0
  nu0     =  0.30d0
  nu1     =  0.30d0
  E0      =  70.0d9
  E1      =  E0*1.0d-09
case (8)
  iss     =  1
  p       =  3
  etadeg  =  68.219578d0
```

```
      Sx      = -1.0d0
      Sy      =  0.0d0
      nu0     =  0.33d0
      nu1     =  0.29d0
      E0      =  70.0d9
      E1      =  E0*3.0d0
  end select

  return
  end subroutine

  !=====================================================================

  subroutine CheckWilsonCoefficients(lu)

  ! For each of the Cases studied in Wilson (1964), recalculate the
  ! series coefficients to check the present programming of this
  ! subsection of the overall solution procedure.
  !
  ! The series coefficents An depend on the value of eta, the specified
  ! material properties of the insert and the plate, as well as the
  ! specified loading.

  implicit none

  integer lu

  integer caseno,p,iss
  real*8  Sx,Sy,nu0,nu1,E0,E1,etadeg

  do caseno = 1, 8
    call GetWilsonData(caseno,p,iss,etadeg,Sx,Sy,nu0,nu1,E0,E1)
    call ComputeShowWilson(lu,caseno,p,iss,etadeg,Sx,Sy,nu0,nu1,E0,E1)
  end do

  return
  end subroutine

  !=====================================================================

  real*8 function FuncNtt(ThetaDeg,EtaDeg,p,A)

  implicit none

  integer p
  real*8  ThetaDeg,EtaDeg,A(0:p)

  real*8  Theta,Eta,Pi,S,Ntt
  integer i

  Pi = 4.0d0*atan(1.0d0)

  Theta = ThetaDeg/180.0d0*Pi
  Eta   = EtaDeg/180.0d0*Pi

  S = 0.0d0
  do I = 0,p
    S = S + A(i)*cos((2.0d0*i+1)*Theta)
  end do

  if (ThetaDeg >= EtaDeg) then
    Ntt = 0.0d0
  else
    Ntt = -8.0d0*sqrt(cos(Theta)**2 - cos(Eta)**2)*S
  end if

  FuncNtt = Ntt
```

```
return
end function

!=====================================================================

real*8 function FuncB(EtaDeg,p,A)

implicit none

integer p
real*8  EtaDeg, A(0:p)

real*8  Delta(0:p), D(0:p+1), Dstar(1:p+1)
real*8  Eta, Cos2Eta, Pi, B
integer i

real*8  Pleg

Pi = 4.0d0*atan(1.0d0)

Eta = EtaDeg/180.0d0*Pi

Delta(0) = 1.0d0
do I = 1,p
  Delta(i) = 0.0d0
end do

Cos2Eta = cos(2*Eta)

D(0) = 1.0d0
D(1) = -Cos2Eta
do I = 2,p+1
  D(i) = (Cos2Eta*Pleg(i-1,Cos2Eta) - Pleg(I,Cos2Eta))/(i-1)
end do

do I = 1,p+1
  Dstar(i) = Delta(i-1) + D(i)
end do

B = 0.0d0
do I = 0,p
  B = B + A(i)*Dstar(i+1)
end do

FuncB = B

return
end function

!=====================================================================

real*8 function FuncStt(ThetaDeg, EtaDeg, Sx, Sy, p, A)

integer p
real*8  ThetaDeg, EtaDeg, Sx, Sy, A(0:p)

real*8  Theta, pi
real*8  Ntt, Stt, B

real*8 FuncNtt, FuncB

Ntt = FuncNtt(ThetaDeg, EtaDeg, p, A)

Pi = 4.0d0*atan(1.0d0)

Theta = ThetaDeg/180.0d0*pi

B = FuncB(EtaDeg,p,A)
```

```
    Stt = Ntt + 4.0d0*B + (Sy+Sx) + 2.0d0*(Sy-Sx)*cos(2.0d0*Theta)

    FuncStt = Stt

    return
    end function

    !=====================================================================

    subroutine GetA0toAp(p,iss,eta,Sx,Sy,nu0,nu1,E0,E1,A,rcond)

    !DEC$ DEFINE UseLAPACK = 1   ! Set to 1 if using LAPACK library.

    implicit none

    integer p,iss
    real*8  eta,Sx,Sy,nu0,nu1,E0,E1,A(0:p),rcond

    real*8  zero,MtxLHS(0:p,0:p),VecRHS(0:p)
    integer I,j,n,k,numint,ipiv(0:p)

    !DEC$ IF (UseLAPACK == 1)
    real*8  work(4*(p+1)),anorm,s
    integer info,iwork(p+1)
    !DEC$ ELSE
    real*8  EigVal(0:p),EigVec(0:p,0:p),TL,MaxEigVal,MinEigVal
    !DEC$ ENDIF

    real*8  cmn_eta,cmn_Sx,cmn_Sy,cmn_nu0,cmn_nu1,cmn_E0,cmn_E1
    integer cmn_n,cmn_k,cmn_iss

    common /cmn_SimEqns/ cmn_eta,cmn_Sx,cmn_Sy,cmn_nu0,cmn_nu1, &
                         cmn_E0,cmn_E1,cmn_n,cmn_k,cmn_iss

    external FuncRHS
    external FuncLHS
    real*8   FuncRHS
    real*8   FuncLHS

    if (eta <= 0.0d0) then
      write(*,*)
      write(*,*) '*** ERROR: Eta (radians) = ',eta
      write(*,*) '***'
      write(*,*) '*** Eta must be greater than zero.'
      write(*,*)
      stop
    end if

    cmn_iss = iss
    cmn_eta = eta
    cmn_Sx  = Sx
    cmn_Sy  = Sy
    cmn_nu0 = nu0
    cmn_nu1 = nu1
    cmn_E0  = E0
    cmn_E1  = E1

    ! Zero the elements of the LHS matrix and RHS vector.

    do k = 0,p
      VecRHS(k) = zero
      do n = 0,p
        MtxLHS(k,n) = zero
      end do
    end do

    ! Set up the system of simultaneous equations defined in Equation (34).
```

```
! Utilise symmetry when computing the matrix of coefficients.

numint = 400

do k = 0,p
  cmn_k = k
  zero  = 0.0d0
  call SimpsonsRule(zero,eta,FuncRHS,numint,VecRHS(k))
  do n = k,p
    cmn_n = n
    call SimpsonsRule(zero,eta,FuncLHS,numint,MtxLHS(k,n))
    MtxLHS(n,k) = MtxLHS(k,n)
  end do
end do

! Solve the system of simultaneous equations.

!DEC$ IF (UseLAPACK == 1)
  !DEC$ MESSAGE:'Solving simultaneous equations using LAPACK library.'
  ! The following procedures assume that the matrix is symmetric.
  call dsytrf('U',p+1,MtxLHS,p+1,ipiv,work,p+1,info)
  anorm = 0.0d0
  do j=0,p
    s = 0.0d0
    do i=0,p
      s = s + abs(MtxLHS(I,j))
    end do
    anorm = max(s,anorm)
  end do
  call dsycon('U',p+1,MtxLHS,p+1,ipiv,anorm,rcond,work,iwork,info)
  call dsytrs('U',p+1,1,MtxLHS,p+1,ipiv,VecRHS,p+1,info)
!DEC$ ELSE
  !DEC$ MESSAGE:'Solving simultaneous equations using DECOMP and SOLVE.'
  TL = 20.0d0
  call JACOBI(MtxLHS,EigVal,EigVec,p+1,TL)
  MaxEigVal = EigVal(0)
  MinEigVal = EigVal(0)
  do I = 1,p
    MaxEigVal = max(EigVal(i),MaxEigVal)
    MinEigVal = min(EigVal(i),MinEigVal)
  end do
  if (MaxEigVal > 0.0d0) then
    rcond = MinEigVal/MaxEigVal
  else
    rcond = 0.0d0
  end if
  call DECOMP(p+1,p+1,MtxLHS,ipiv)
  call SOLVE(p+1,p+1,MtxLHS,VecRHS,ipiv)
!DEC$ ENDIF

do k = 0,p
  A(k) = VecRHS(k)
end do

return
end subroutine

!=======================================================================

real*8 function FuncRHS(theta)

real*8  theta

real*8  eta,Sx,Sy,nu0,nu1,E0,E1
integer n,k,iss

common /cmn_SimEqns/ eta,Sx,Sy,nu0,nu1,E0,E1,n,k,iss
```

```
    real*8  Gntheta

    FuncRHS = -((Sy+Sx)+6.0d0*(Sy-Sx)*cos(2.0d0*theta))* &
              Gntheta(k,theta,eta,iss,nu0,nu1,E0,E1)

    return
    end function

    !=====================================================================

    real*8 function FuncLHS(theta)

    real*8  theta

    real*8  eta,Sx,Sy,nu0,nu1,E0,E1
    integer n,k,iss

    common /cmn_SimEqns/ eta,Sx,Sy,nu0,nu1,E0,E1,n,k,iss

    real*8  Gntheta

    FuncLHS = Gntheta(n,theta,eta,iss,nu0,nu1,E0,E1)* &
              Gntheta(k,theta,eta,iss,nu0,nu1,E0,E1)

    return
    end function

    !=====================================================================

    real*8 function Dstarnp1(n,eta)

    implicit none

    integer n
    real*8  eta

    real*8  Dketa

    if (n == 0) then
      Dstarnp1 = 1.0d0 + Dketa(n+1,eta)
    else
      Dstarnp1 = 0.0d0 + Dketa(n+1,eta)
    end if

    return
    end function

    !=====================================================================

    real*8 function Gntheta(n,theta,eta,iss,nu0,nu1,E0,E1)

    implicit none

    integer n
    integer iss
    real*8  theta,eta,nu0,nu1,E0,E1

    real*8  k1,k2,k3,Nn,In,Ds

    real*8  Nntheta,Intheta,Dstarnp1

    call Computek1k2k3(k1,k2,k3,iss,nu0,nu1,E0,E1)

    Nn = Nntheta(n,theta,eta)
    In = Intheta(n,theta,eta)
    Ds = Dstarnp1(n,eta)

    Gntheta = k1*Nn + k2*In + 4.0d0*Ds
```

```
return
end function

!=====================================================================

subroutine SimpsonsRule(a, b, f, neven, v)

! Integration of a function f(x) over the interval [a, b] using
! Simpson's Rule with neven intervals (neven is an even number).
! The result is returned in v.

implicit none

real*8   a,b,v
integer  neven
external f
real*8   f

integer  i
real*8   r,s

s = (b-a)/dfloat(neven)
v = (f(a)+f(b))/2.0d0

do I = 1,neven-1
  r = f(a+i*s)
  if (mod(I,2).ne.0) then
    v = v + 2.0d0*r
  else
    v = v + r
  end if
end do

v = v*s*2.0d0/3.0d0

return
end subroutine

!=====================================================================

real*8 function FuncEnIntegral(theta)

implicit none

real*8  theta

real*8  eta
integer n

common /cmn_FuncEnIntegral/ eta, n

FuncEnIntegral = sin(theta)*cos((2*n+1)*theta)* &
                 sqrt(abs(cos(theta)**2-cos(eta)**2))

return
end function

!=====================================================================

real*8 function Eneta(n,eta)

implicit none

integer  n
real*8   eta,zero

real*8   EnIntegral
```

```
    integer  numint

    external FuncEnIntegral
    real*8   FuncEnIntegral

    real*8   cmn_eta
    integer  cmn_n

    common /cmn_FuncEnIntegral/ cmn_eta, cmn_n

    cmn_n   = n
    cmn_eta = eta

    numint = 400
    zero   = 0.0d0

    call SimpsonsRule(zero,eta,FuncEnIntegral,numint,EnIntegral)

    Eneta = -8.0d0*EnIntegral

    return
    end function

    !=====================================================================

    subroutine Test_FuncFnIntegral

    implicit none

    real*8  eta,theta
    integer n,i

    common /cmn_FuncFnIntegral/ eta, n

    real*8   FuncFnIntegral

    eta = 0.10d0
    n   = 4

    write(*,'(2f25.20)') 0.0d+00,FuncFnIntegral(0.0d+00)
    write(*,'(2f25.20)') 1.0d-15,FuncFnIntegral(1.0d-15)
    write(*,'(2f25.20)') 1.0d-14,FuncFnIntegral(1.0d-14)
    write(*,'(2f25.20)') 1.0d-13,FuncFnIntegral(1.0d-13)
    write(*,'(2f25.20)') 1.0d-12,FuncFnIntegral(1.0d-12)
    write(*,'(2f25.20)') 1.0d-11,FuncFnIntegral(1.0d-11)
    write(*,'(2f25.20)') 1.0d-10,FuncFnIntegral(1.0d-10)
    write(*,'(2f25.20)') 1.0d-09,FuncFnIntegral(1.0d-09)
    write(*,'(2f25.20)') 1.0d-08,FuncFnIntegral(1.0d-08)
    write(*,'(2f25.20)') 1.0d-07,FuncFnIntegral(1.0d-07)
    do I = 1,10
      theta = 0.000001d0*i
      write(*,'(2f25.20)') theta,FuncFnIntegral(theta)
    end do
    do I = 1,100
      theta = eta*i/100
      write(*,'(2f25.20)') theta,FuncFnIntegral(theta)
    end do

    stop
    end subroutine

    !=====================================================================

    real*8 function FuncFnIntegral(theta)

    implicit none

    real*8  theta
```

```
real*8  eta
integer n

common /cmn_FuncFnIntegral/ eta, n

if (theta < 1.0d-09) then
  ! Special result that is applicable at theta = 0.
  FuncFnIntegral = sin(eta)*log(2.0d0)
else
  FuncFnIntegral = cos(theta)*cos((2*n+1)*theta)*         &
                   log(cotan(theta/2))*                   &
                   sqrt(abs(cos(theta)**2-cos(eta)**2)) + &
                   sin(eta)*log(theta)
end if

return
end function

!=====================================================================

real*8 function Fneta(n,eta)

implicit none

integer  n
real*8   eta

real*8   zero,pi,FnIntegral
integer  numint

external FuncFnIntegral
real*8   FuncFnIntegral

real*8   cmn_eta
integer  cmn_n

common /cmn_FuncFnIntegral/ cmn_eta, cmn_n

cmn_n   = n
cmn_eta = eta

numint = 400
zero = 0.0d0

call SimpsonsRule(zero,eta,FuncFnIntegral,numint,FnIntegral)

pi   = 4.0d0*atan(1.0d0)

Fneta = 16.0d0/pi*(FnIntegral + eta*sin(eta)*(1.0d0-log(eta)))

return
end function

!=====================================================================

real*8 function Nntheta(n,theta,eta)

implicit none

real*8  theta,eta
integer n

Nntheta = -8.0d0*cos((2*n+1)*theta)*sqrt(abs(cos(theta)**2-cos(eta)**2))

return
end function
```

```
!=====================================================================

real*8 function Intheta(n,theta,eta)

implicit none

real*8  theta,eta
integer n

integer k
real*8  s

real*8  Dketa

s = 0.0d0
do k = 0,n
  s = s + (n+1-k)*Dketa(k,eta)*cos(2*(n+1-k)*theta)
end do

Intheta = -8.0d0*s

return
end function

!=====================================================================

real*8 function Dketa(k,eta)

implicit none

integer k
real*8  eta

real*8  Pleg

if (k == 0) then
  Dketa = 1
else if (k == 1) then
  Dketa = -cos(2.0d0*eta)
else
  Dketa = ( cos(2.0d0*eta)*Pleg(k-1,cos(2.0d0*eta)) - &
            Pleg(k,cos(2.0d0*eta))                   )/(k-1)
end if

return
end function

!=====================================================================

subroutine ComputeWilsonCases1to8(lu,nOut)

implicit none

integer lu,nOut

integer CaseNo,i
real*8  A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0,nu0,nu1
real*8  ThetaDeg,Stt,Ntt,thetanext

real*8  Wilson1964_Stt
real*8  Wilson1964_Ntt

do CaseNo = 1,8

  call WilsonCaseData(CaseNo,A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0,nu0,nu1)

  write(lu,*)
  write(lu,'(a,i1   )') 'RESULTS USING DATA FROM WILSON''S CASE ',CaseNo
```

```
  write(lu,'(a      )') '======================================'
  write(lu,*)
  write(lu,'(a,f14.6)') 'Sy        = ',Sy
  write(lu,'(a,f14.6)') 'Sx        = ',Sx
  write(lu,'(a,f14.6)') 'E1/E0     = ',E1onE0
  if (nu0 > 0.0d0) then
    write(lu,'(a,f14.6)') 'nu0       = ',nu0
  else
    write(lu,'(a,a14  )') 'nu0       = ','Arbitrary'
  endif
  write(lu,'(a,f14.6)') 'nu1       = ',nu1
  write(lu,*)
  write(lu,'(a,f14.6)') 'Eta (deg) = ',EtaDeg
  write(lu,*)
  write(lu,'(a,f14.6)') 'A0        = ',A0
  write(lu,'(a,f14.6)') 'A1        = ',A1
  write(lu,'(a,f14.6)') 'A2        = ',A2
  write(lu,'(a,f14.6)') 'A3        = ',A3
  write(lu,'(a,f14.6)') 'A4        = ',A4
  write(lu,*)
  write(lu,'(a,f14.6)') 'Ntt(0)    = ',Wilson1964_Ntt( 0.0d0,CaseNo)
  write(lu,'(a,f14.6)') 'Stt(0)    = ',Wilson1964_Stt( 0.0d0,CaseNo)
  write(lu,'(a,f14.6)') 'Stt(Eta)  = ',Wilson1964_Stt(EtaDeg,CaseNo)
  write(lu,'(a,f14.6)') 'Stt(90)   = ',Wilson1964_Stt(90.0d0,CaseNo)

  write(lu,*)
  write(lu,'(3a15)') 'Theta_deg','Stt','Ntt'
  do I = 0,nOut
    ThetaDeg = 90.0d0*i/nOut
    Stt = Wilson1964_Stt(ThetaDeg, CaseNo)
    Ntt = Wilson1964_Ntt(ThetaDeg, CaseNo)
    write(lu,'(f15.6,2f15.9)') ThetaDeg,Stt,Ntt
    thetanext = (i+1)*90.0d0/nOut
    if (EtaDeg > Thetadeg .and. EtaDeg < thetanext) then
      Stt = Wilson1964_Stt(EtaDeg, CaseNo)
      Ntt = Wilson1964_Ntt(EtaDeg, CaseNo)
      write(lu,'(f15.6,2f15.8)') EtaDeg, Stt, Ntt
    end if
  end do

end do

return
end subroutine

!=======================================================================

subroutine WilsonCaseData(CaseNo,A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0, &
                          nu0,nu1)

implicit none

integer CaseNo
real*8  A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0,nu0,nu1

select case (CaseNo)
  case (1)
    EtaDeg = 17.285975d0
    A0     =  0.485594d0
    A1     = -0.032165d0
    A2     =  0.008132d0
    A3     = -0.001099d0
    A4     =  0.0d0
    Sx     =  0.0d0
    Sy     =  1.0d0
    E1onE0 =  0.0d0
    nu0    =  0.00d0
    nu1    =  0.50d0
```

```
      case (2)
        EtaDeg = 19.625247d0
        A0     =  0.238485d0
        A1     = -0.015260d0
        A2     =  0.004035d0
        A3     = -0.000575d0
        A4     =  0.0d0
        Sx     =  0.0d0
        Sy     =  1.0d0
        E1onE0 =  1.0d0
        nu0    =  0.30d0
        nu1    =  0.30d0
      case (3)
        EtaDeg = 18.338752d0
        A0     =  0.720664d0
        A1     = -0.492358d0
        A2     =  0.231672d0
        A3     = -0.026564d0
        A4     = -0.007271d0
        Sx     =  0.0d0
        Sy     =  1.0d0
        E1onE0 =  0.0d0
        nu0    =  0.00d0
        nu1    =  0.30d0
      case (4)
        EtaDeg = 20.580669d0
        A0     =  0.100991d0
        A1     =  0.034317d0
        A2     = -0.021163d0
        A3     =  0.004578d0
        A4     =  0.0d0
        Sx     =  0.0d0
        Sy     =  1.0d0
        E1onE0 =  3.0d0
        nu0    =  0.33d0
        nu1    =  0.29d0
      case (5)
        EtaDeg = 40.472075d0
        A0     =  0.410995d0
        A1     = -0.012116d0
        A2     =  0.004066d0
        A3     = -0.001023d0
        A4     =  0.000140d0
        Sx     = -1.0d0
        Sy     =  0.0d0
        E1onE0 =  0.0d0
        nu0    =  0.00d0
        nu1    =  0.50d0
      case (6)
        EtaDeg = 56.973112d0
        A0     =  0.192362d0
        A1     = -0.001886d0
        A2     =  0.000772d0
        A3     = -0.000261d0
        A4     =  0.000055d0
        Sx     = -1.0d0
        Sy     =  0.0d0
        E1onE0 =  1.0d0
        nu0    =  0.30d0
        nu1    =  0.30d0
      case (7)
        EtaDeg = 49.813774d0
        A0     =  0.352945d0
        A1     = -0.015537d0
        A2     =  0.008636d0
        A3     = -0.003458d0
        A4     =  0.000763d0
        Sx     = -1.0d0
```

```
    Sy     =  0.0d0
    E1onE0 =  0.0d0
    nu0    =  0.00d0
    nu1    =  0.30d0
  case (8)
    EtaDeg = 68.219578d0
    A0     =  0.101956d0
    A1     = -0.000158d0
    A2     =  0.000050d0
    A3     = -0.000011d0
    A4     =  0.0d0
    Sx     = -1.0d0
    Sy     =  0.0d0
    E1onE0 =  3.0d0
    nu0    =  0.33d0
    nu1    =  0.29d0
end select

return
end subroutine

!=======================================================================

real*8 function Wilson1964_B(CaseNo)

implicit none

integer CaseNo

real*8  Delta00, Delta01, Delta02, Delta03, Delta04
real*8  A0, A1, A2, A3, A4
real*8  B0, B1, B2, B3, B4
real*8  D0, D1, D2, D3, D4, D5
real*8  Dstar1, Dstar2, Dstar3, Dstar4, Dstar5
real*8  EtaDeg, Eta, Cos2Eta, Pi, Sx, Sy, E1onE0, nu0, nu1

real*8  Pleg

call WilsonCaseData(CaseNo,A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0,nu0,nu1)

Pi = 4.0d0*atan(1.0d0)

Eta = EtaDeg/180.0d0*Pi

Delta00 = 1.0d0
Delta01 = 0.0d0
Delta02 = 0.0d0
Delta03 = 0.0d0
Delta04 = 0.0d0

Cos2Eta = cos(2*Eta)

D0 = 1.0d0
D1 = -Cos2Eta
D2 = (Cos2Eta*Pleg(2-1,Cos2Eta) - Pleg(2,Cos2Eta))/(2-1)
D3 = (Cos2Eta*Pleg(3-1,Cos2Eta) - Pleg(3,Cos2Eta))/(3-1)
D4 = (Cos2Eta*Pleg(4-1,Cos2Eta) - Pleg(4,Cos2Eta))/(4-1)
D5 = (Cos2Eta*Pleg(5-1,Cos2Eta) - Pleg(5,Cos2Eta))/(5-1)

Dstar1 = Delta00 + D1
Dstar2 = Delta01 + D2
Dstar3 = Delta02 + D3
Dstar4 = Delta03 + D4
Dstar5 = Delta04 + D5

B0 = A0*Dstar1
B1 = A1*Dstar2
B2 = A2*Dstar3
```

```
B3 = A3*Dstar4
B4 = A4*Dstar5

Wilson1964_B = B0 + B1 + B2 + B3 + B4

return
end function

!=====================================================================

real*8 function Wilson1964_Ntt(ThetaDeg,CaseNo)

implicit none

real*8  ThetaDeg
integer CaseNo

real*8  Theta, EtaDeg, Eta, Pi
real*8  A0, A1, A2, A3, A4
real*8  S, Ntt, Sx, Sy, E1onE0, nu0, nu1

call WilsonCaseData(CaseNo,A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0,nu0,nu1)

Pi = 4.0d0*atan(1.0d0)

Theta = ThetaDeg/180.0d0*Pi
Eta   = EtaDeg/180.0d0*Pi

S = 0.0d0
S = S + A0*cos((2*0+1)*Theta)
S = S + A1*cos((2*1+1)*Theta)
S = S + A2*cos((2*2+1)*Theta)
S = S + A3*cos((2*3+1)*Theta)
S = S + A4*cos((2*4+1)*Theta)

if (ThetaDeg >= EtaDeg) then
  Ntt = 0.0d0
else
  Ntt = -8.0d0*sqrt(cos(Theta)**2 - cos(Eta)**2)*S
end if

Wilson1964_Ntt = Ntt

return
end function

!=====================================================================

real*8 function Wilson1964_Stt(ThetaDeg, CaseNo)

real*8  ThetaDeg
integer CaseNo

real*8  EtaDeg, Theta, Pi
real*8  A0, A1, A2, A3, A4
real*8  Ntt, Stt, B
real*8  Sx, Sy, E1onE0, nu0, nu1

real*8 Wilson1964_Ntt, Wilson1964_B

call WilsonCaseData(CaseNo,A0,A1,A2,A3,A4,EtaDeg,Sx,Sy,E1onE0,nu0,nu1)

Ntt = Wilson1964_Ntt(ThetaDeg, CaseNo)

Pi = 4.0d0*atan(1.0d0)

Theta = ThetaDeg/180.0d0*Pi
```

```
B = Wilson1964_B(CaseNo)

Stt = Ntt + 4.0d0*B + (Sy+Sx) + 2.0d0*(Sy-Sx)*cos(2*Theta)

Wilson1964_Stt = Stt

return
end function

!=====================================================================

real*8 function Pleg(n,x)

! Calculate the value of the Legendre polynomial Pn(x).
!
! Parameters:
!
!    n = degree of polynomial, n >= 0
!    x = argument
!
! Result:
!
!    The value of the Legendre polynomial Pn at x
!
! Some test values taken from Abramowitz and Stegun are shown in {}.
!
! Pleg(0,0.3141592654d0) =  1.0000000000   { 1.0000000000}
! Pleg(1,0.3141592654d0) =  0.3141592654   { 0.3141592654}
! Pleg(2,0.3141592654d0) = -0.3519559339   {-0.3519559340}
! Pleg(3,0.3141592654d0) = -0.3937232064   {-0.3937232064}
! Pleg(4,0.3141592654d0) =  0.0475063122   { 0.0475063122}
! Pleg(5,0.3141592654d0) =  0.3418427518   { 0.3418427517}
! Pleg(6,0.3141592654d0) =  0.1572986974   { 0.1572986975}
! Pleg(7,0.3141592654d0) = -0.2012339355   {-0.2012339354}
! Pleg(8,0.3141592654d0) = -0.2561729328   {-0.2561729328}
!
! Milton Abramowitz, Irene Stegun. Handbook of Mathematical Functions
! With Formulas, Graphs, and Mathematical Tables. US Department of
! Commerce, National Bureau of Standards, Applied Mathematics Series 55,
! Issued 1964, Fifth Printing, August 1966, with corrections.

implicit none

integer n
real*8  x

real*8  Rslt, a, b
integer i

a = 1.0d0
b = x

if (n == 0) then
  Pleg = a
  return
end if

if (n == 1) then
  Pleg = b
  return
end if

do I = 2, n
  Rslt = ((2*i-1)*x*b - (i-1)*a)/i
  a = b
  b = Rslt
end do
```

```
    Pleg = Rslt

    return
    end function

    !=====================================================================

    SUBROUTINE DECOMP(N, NDIM, A, IP)

    ! TOMS ALGORITHM 423 - MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION
    !
    ! TOMS 423 is a FORTRAN 77 program that implements ACM TOMS Algorithm
    ! 423, for Gaussian elimination to factor a matrix and solve a related
    ! linear system.
    !
    ! The algorithm was originally written using single precision
    ! arithmetic, but has been converted to double precision here.
    !
    ! Source code is a modification of Algorithm 423: 'Linear Equation
    ! Solver' by Cleve B. Moler, Communications of the ACM, April 1972,
    ! Volume 15, Number 4.
    !
    ! Input:
    !
    !   N    = order of matrix.
    !   NDIM = declared dimension of array A.
    !   A    = matrix to be triangularized.
    !
    ! Output:
    !
    !   A(I,J), I.LE.J = upper triangular factor U.
    !   A(I,J), I.GT.J = multipliers = lower triangular factor, I-L.
    !   IP(K),  K.LT.N = index of K-th pivot row.
    !   IP(N)          = (-1)**(number of interchanges) or 0.
    !
    ! Use SOLVE to obtain the solution of the linear system of
    ! simultaneous equations.
    !
    ! Note that DETERMINANT(A) = IP(N)*A(1,1)*A(2,2)*...*A(N,N).
    !
    ! If IP(N) = 0, then A is singular, and SOLVE will divide by zero.
    ! Interchanges finished in U, only partly in L.

    IMPLICIT NONE

    INTEGER N, NDIM, IP(NDIM)
    real*8  A(NDIM,NDIM)

    real*8  T
    INTEGER I, J, K, KP1, M

      IP(N) = 1
      DO 6 K = 1, N
        IF ( K .EQ. N ) GO TO 5
        KP1 = K + 1
        M = K
        DO 1 I = KP1, N
          IF ( ABS(A(I,K)) .GT. ABS(A(M,K)) ) M = I
    1   CONTINUE
        IP(K) = M
        IF ( M .NE. K ) IP(N) = -IP(N)
        T = A(M,K)
        A(M,K) = A(K,K)
        A(K,K) = T
        IF ( T .EQ. 0.0D0 ) GO TO 5
        DO 2 I = KP1, N
    2     A(I,K) = -A(I,K) / T
        DO 4 J = KP1, N
```

```
      T = A(M,J)
      A(M,J) = A(K,J)
      A(K,J) = T
      IF ( T .EQ. 0.0D0 ) GO TO 4
      DO 3 I = KP1, N
3       A(I,J) = A(I,J) + A(I,K) * T
4   CONTINUE
5   IF ( A(K,K) .EQ. 0.0D0 ) IP(N) = 0
6 CONTINUE

RETURN
END SUBROUTINE

!=====================================================================

SUBROUTINE SOLVE(N, NDIM, A, B, IP)

! Solution of linear system of equations, A*X = B.
!
! Input:
!
!   N    = order of matrix.
!   NDIM = declared dimension of array A.
!   A    = triangularized matrix obtained from DECOMP.
!   B    = right hand side vector.
!   IP   = pivot vector obtained from DECOMP.
!
! Do not use results if DECOMP has set IP(N) = 0.
!
! Output:
!
!   B = solution vector, X.

IMPLICIT NONE

INTEGER N, NDIM, IP(NDIM)
real*8  A(NDIM,NDIM), B(NDIM)

real*8  T
INTEGER KM1, KP1, NM1, I, K, KB, M

  IF ( N .EQ. 1 ) GO TO 9
  NM1 = N - 1
  DO 7 K = 1, NM1
    KP1 = K + 1
    M = IP(K)
    T = B(M)
    B(M) = B(K)
    B(K) = T
    DO 7 I = KP1, N
7     B(I) = B(I) + A(I,K) * T
  DO 8 KB = 1, NM1
    KM1 = N - KB
    K = KM1 + 1
    B(K) = B(K) / A(K,K)
    T = -B(K)
    DO 8 I = 1,KM1
8     B(I) = B(I) + A(I,K) * T
9   B(1) = B(1) / A(1,1)

RETURN
END SUBROUTINE

!=====================================================================

subroutine TestDFZERO

implicit none
```

```
real*8  b, c, r, re, ae
integer iflag

external TestFuncDFZERO
real*8   TestFuncDFZERO

b  =  0.0d0
c  = 90.0d0
r  = b
re = 1.0d-07
ae = 0.0d0

call DFZERO(TestFuncDFZERO, b, c, r, re, ae, iflag)

write(*,'(a,i20)'   ) 'IFLAG        = ',iflag
write(*,'(a,f20.10)') 'Zero of F(x) = ',b

stop

return
end subroutine

!======================================================================

real*8 function TestFuncDFZERO(x)

implicit none

real*8 x

TestFuncDFZERO = sind(x)-1.0d0/sqrt(2.0d0) ! F(x) = 0 when x = 45.

return
end function

!======================================================================

SUBROUTINE DFZERO(F, B, C, R, RE, AE, IFLAG)

!***BEGIN PROLOGUE  DFZERO
!
!***PURPOSE
!
! Search for a zero of a function F(X) in a given interval (B,C).
! It is designed primarily for problems where F(B) and F(C) have
! opposite signs.
!
!***LIBRARY   SLATEC
!***CATEGORY  F1B
!***TYPE      DOUBLE PRECISION (FZERO-S, DFZERO-D)
!***KEYWORDS  BISECTION, NONLINEAR, ROOTS, ZEROS
!***AUTHORS   L. F. Shampine (SNLA)
!             H. A. Watts    (SNLA)
!
!***DESCRIPTION
!
! DFZERO searches for a zero of a DOUBLE PRECISION function F(X)
! between the given DOUBLE PRECISION values B and C until the width
! of the interval (B,C) has collapsed to within a tolerance
! specified by the stopping criterion,
!
!    ABS(B-C) .LE. 2.*(RW*ABS(B)+AE).
!
! The method used is an efficient combination of bisection and the
! secant rule, and is due to T. J. Dekker.
!
! Description Of Arguments
```

```
!
!   F      :EXT   - Name of the DOUBLE PRECISION external function. This
!                   name must be in an EXTERNAL statement in the calling
!                   program. F must be a function of one DOUBLE
!                   PRECISION argument.
!
!   B      :INOUT - One end of the DOUBLE PRECISION interval (B,C). The
!                   value returned for B usually is the better
!                   approximation to a zero of F.
!
!   C      :INOUT - The other end of the DOUBLE PRECISION interval (B,C)
!
!   R      :IN    - A (better) DOUBLE PRECISION guess of a zero of F
!                   which could help in speeding up convergence. If F(B)
!                   and FI have opposite signs, a root will be found in
!                   the interval (B,R); if not, but FI and F(C) have
!                   opposite signs, a root will be found in the interval
!                   (R,C); otherwise, the interval (B,C) will be
!                   searched for a possible root. When no better guess
!                   is known, it is recommended that R be set to B or C,
!                   since if R is not interior to the interval (B,C), it
!                   will be ignored.
!
!   RE     :IN    - Relative error used for RW in the stopping criterion.
!                   If the requested RE is less than machine precision,
!                   then RW is set to approximately machine precision.
!
!   AE     :IN    - Absolute error used in the stopping criterion. If the
!                   given interval (B,C) contains the origin, then a
!                   nonzero value should be chosen for AE.
!
!   IFLAG :OUT    - A status code. User must check IFLAG after each call.
!                   Control returns to the user from DFZERO in all cases.
!
!                     1  B is within the requested tolerance of a zero.
!                        The interval (B,C) collapsed to the requested
!                        tolerance, the function changes sign in (B,C), and
!                        F(X) decreased in magnitude as (B,C) collapsed.
!
!                     2  F(B) = 0. However, the interval (B,C) may not have
!                        collapsed to the requested tolerance.
!
!                     3  B may be near a singular point of F(X). The
!                        interval (B,C) collapsed to the requested
!                        tolerance and the function changes sign in
!                        (B,C), but F(X) increased in magnitude as
!                        (B,C) collapsed, i.e.
!                        ABS(F(B out)) .GT. MAX(ABS(F(B in)),ABS(F(C in)))
!
!                     4  No change in sign of F(X) was found although the
!                        interval (B,C) collapsed to the requested
!                        tolerance. The user must examine this case and
!                        decide whether B is near a local minimum of F(X),
!                        or B is near a zero of even multiplicity, or
!                        neither of these.
!
!                     5  Too many (.GT. 1000) function evaluations used.
!
!***REFERENCES
!
!  L. F. Shampine and H. A. Watts. FZERO, a root-solving code.
!  Report SC-TM-70-631, Sandia Laboratories, September 1970.
!
!  T. J. Dekker. Finding a zero by means of successive
!  linear interpolation. Constructive Aspects of the
!  Fundamental Theorem of Algebra, edited by B. Dejon
!  and P. Henrici, Wiley-Interscience, 1969.
!
```

```
!***ROUTINES CALLED  D1MACH
!
!***REVISION HISTORY  (YYMMDD)
!
!   700901  DATE WRITTEN
!   890531  Changed all specific intrinsics to generic.  (WRB)
!   890531  REVISION DATE from Version 3.2
!   891214  Prologue converted to Version 4.0 format.  (BAB)
!   920501  Reformatted the REFERENCES section.  (WRB)
!
!***END PROLOGUE  DFZERO

   real*8  A,ACBS,ACMB,AE,AW,B,C,CMB,ER
   real*8  F,FA,FB,FC,FX,FZ,P,Q,R,RE,RW,T,TOL,Z
   INTEGER IC,IFLAG,KOUNT

   !***FIRST EXECUTABLE STATEMENT  DFZERO

   ! ER is two times the computer unit roundoff value which is defined
   ! here by the function D1MACH.

   ! ER = 2.0D0 * D1MACH(4)

   ER = 2.0D0*1.0D-14

   ! Initialize.

   Z = R
   IF (R .LE. MIN(B,C)  .OR.  R .GE. MAX(B,C)) Z = C
   RW = MAX(RE,ER)
   AW = MAX(AE,0.D0)
   IC = 0
   T = Z
   FZ = F(T)
   FC = FZ
   T = B
   FB = F(T)
   KOUNT = 2
   IF (SIGN(1.0D0,FZ) .EQ. SIGN(1.0D0,FB)) GO TO 1
   C = Z
   GO TO 2
 1 IF (Z .EQ. C) GO TO 2
   T = C
   FC = F(T)
   KOUNT = 3
   IF (SIGN(1.0D0,FZ) .EQ. SIGN(1.0D0,FC)) GO TO 2
   B = Z
   FB = FZ
 2 A = C
   FA = FC
   ACBS = ABS(B-C)
   FX = MAX(ABS(FB),ABS(FC))

 3 IF (ABS(FC) .GE. ABS(FB)) GO TO 4

  ! Perform interchange.

   A = B
   FA = FB
   B = C
   FB = FC
   C = A
   FC = FA

 4 CMB = 0.5D0*(C-B)
   ACMB = ABS(CMB)
   TOL = RW*ABS(B) + AW
```

```
   ! Test stopping criterion and function count.

   IF (ACMB .LE. TOL) GO TO 10
   IF (FB .EQ. 0.D0) GO TO 11
   IF (KOUNT .GE. 1000) GO TO 14

   ! Calculate new iterate implicitly as B+P/Q, where we arrange
   ! P .GE. 0.  The implicit form is used to prevent overflow.

   P = (B-A)*FB
   Q = FA - FB
   IF (P .GE. 0.D0) GO TO 5
   P = -P
   Q = -Q

   ! Update A and check for satisfactory reduction in the size of the
   ! bracketing interval.  If not, perform bisection.

 5 A = B
   FA = FB
   IC = IC + 1
   IF (IC .LT. 4) GO TO 6
   IF (8.0D0*ACMB .GE. ACBS) GO TO 8
   IC = 0
   ACBS = ACMB

   ! Test for too small a change.

 6 IF (P .GT. ABS(Q)*TOL) GO TO 7

   ! Increment by TOLerance.

   B = B + SIGN(TOL,CMB)
   GO TO 9

   ! Root ought to be between B and (C+B)/2.

 7 IF (P .GE. CMB*Q) GO TO 8

   ! Use secant rule.

   B = B + P/Q
   GO TO 9

   ! Use bisection (C+B)/2.

 8 B = B + CMB

   ! Have completed computation for new iterate B.

 9 T = B
   FB = F(T)
   KOUNT = KOUNT + 1

   ! Decide whether next step is interpolation or extrapolation.

   IF (SIGN(1.0D0,FB) .NE. SIGN(1.0D0,FC)) GO TO 3
   C = A
   FC = FA
   GO TO 3

   ! Finished. Process results for proper setting of IFLAG.

10 IF (SIGN(1.0D0,FB) .EQ. SIGN(1.0D0,FC)) GO TO 13
   IF (ABS(FB) .GT. FX) GO TO 12
   IFLAG = 1
   RETURN
11 IFLAG = 2
```

```
   RETURN
12 IFLAG = 3
   RETURN
13 IFLAG = 4
   RETURN
14 IFLAG = 5
   RETURN

END SUBROUTINE

!=====================================================================

subroutine TestCompleteEllipticIntegrals12

! Test program to check the computation of the complete elliptic
! integrals of the first and second kind.
!
! Values from the tables in Abramowitz and Stegun are marked as “A&S”.
!
!      m                      K(m)                  E(m)
!  0.000     1.570796326794897E+00     1.570796326794897
!  0.050     1.591003453790792E+00     1.550973351780472
!  0.100     1.612441348720219E+00     1.530757636897763
!  0.150     1.635256732264580E+00     1.510121832092820
!  0.200     1.659623598610528E+00     1.489035058095853
!  0.250     1.685750354812596E+00     1.467462209339427
!  0.300     1.713889448178791E+00     1.445363064412665
!  0.300 A&S 1.713889448178791         1.445363064
!  0.350     1.744350597225613E+00     1.422691133490879
!  0.400     1.777519371491253E+00     1.399392138897432
!  0.450     1.813883936816983E+00     1.375401971871116
!  0.500     1.854074677301372E+00     1.350643881047675
!  0.500 A&S 1.854074677301372         1.350643881
!  0.550     1.898924910271554E+00     1.325024497958230
!  0.600     1.949567749806026E+00     1.298428035046913
!  0.650     2.007598398424376E+00     1.270707479650149
!  0.700     2.075363135292469E+00     1.241670567945823
!  0.750     2.156515647499643E+00     1.211056027568459
!  0.750 A&S 2.156515647499643         1.211056028
!  0.800     2.257205326820854E+00     1.178489924327838
!  0.850     2.389016486325580E+00     1.143395791883166
!  0.900     2.578092113348173E+00     1.104774732704073
!  0.950     2.908337248444552E+00     1.060473727766279
!  0.950 A&S 2.908337248444552         1.060473728
!  0.990     3.695637362989874E+00     1.015993545025224
!  0.990 A&S 3.695637362989875         1.015993456
!  1.000     1.000000000000000+300     1.000000000000000
!  1.000 A&S Infinity                  1.000000000

implicit none

real*8  e1,e2,m
integer I, ni

m  = 0.0d0
ni = 100

write(*,’(a6,a26,a22)’) ‘m’,’K(m)’,’E(m)’

do I = 0, ni
  m = i*1.0d0/ni
  call CompleteEllipticIntegrals12(m,e1,e2)
  write(*,’(f6.3,es26.15,f22.15)’) m,e1,e2
end do

stop

end subroutine
```

```
!=====================================================================

subroutine CompleteEllipticIntegrals12(m,EIK,EIE)

! This procedure computes the complete elliptic integrals of the first
! and second kind, K(m) and E(m). The computational technique uses an
! arithmetic-geometric-mean process.
!
! The input parameter is m, where 0 <= m <= 1.
!
! The returned values are EIK and EIE, the complete elliptic integrals
! of the first and second kind, respectively.
!
! Note that m1 = 1 - k^2 = 1 -m = cos(alpha)**2.
!
! The complementary parameter, m1, is chosen as the independent
! variable, rather than the parameter m, the modulus k, or the
! modular angle alpha, because of the possibility of serious
! loss of significance in generating m1 from the other possible
! independent variables when m1 is small and dK/dm1 is very large.
!
! Reference:
!
! Algorithm 165, Complete Elliptic Integrals. H. C. Thatcher.
! Communications of the ACM, Volume 6, Number 4, April 1963,
! pages 163-164.

implicit none

real*8  m,EIK,EIE

real*8  m1,tol,pi,a,b,c,s,temp
integer fact

if (m == 1.0d0) then
  EIK = 1.0d+300
  EIE = 1.0d0
  return
end if

tol = 5.0d-07

pi = 4.0d0*atan(1.0d0)

m1 = 1.0d0 - m

if (m1 < 0.0 .or. m1 > 1.0d0) then
  write(*,*) &
    '*** Error: m1 is out of range in CompleteEllipticIntegrals12.'
  stop
end if

a    = 1.0d0
fact = 1
b    = sqrt(m1)
temp = 1.0d0 - m1
s    = 0.0d0

100 continue

s    = s + temp
c    = (a-b)/2.0d0
fact = fact + fact
temp = (a+b)/2.0d0
b    = sqrt(a*b)
a    = temp
temp = fact*c**2
```

```
    if (abs(c) >= tol*a .or. temp > tol*s) goto 100

    s = s + temp

    EIK = pi/(a+b)
    EIE = EIK*(1.0d0-s/2.0d0)

    return
    end subroutine

    !=====================================================================

    SUBROUTINE JACOBI(SYMMTX,EIGVAL,EIGVEC,NEQ,TL)

    ! EIGENVALUE SOLUTION BY JACOBI METHOD.
    !
    ! ORIGINALLY WRITTEN BY ED WILSON, 25 DECEMBER 1990.
    ! MODIFIED 14 OCTOBER 2010.
    !
    ! NEQ    - ORDER OF THE SQUARE MATRIX.
    ! SYMMTX - MATRIX (ANY RANK) TO BE SOLVED.
    ! A      - WORKING COPY OF INPUT MATRIX TO BE SOLVED.
    !          EIGENVALUES STORED ON THE DIAGONAL.
    ! EIGVEC - MATRIX OF EIGENVECTORS PRODUCED.
    ! EIGVAL - VECTOR COLUMN OF COMPUTED EIGENVALUES.
    ! TL     - NUMBER OF SIGNIFICANT FIGURES.

    IMPLICIT REAL*8 (A-H,O-Z)

    DIMENSION SYMMTX(NEQ,NEQ),EIGVAL(NEQ),EIGVEC(NEQ,NEQ)

    DIMENSION A(NEQ,NEQ)

    !---- INITIALIZATION -----------------------
    DO I=1,NEQ
      DO J=1,NEQ
        A(I,J) = SYMMTX(I,J)
      END DO
    END DO
    ZERO = 0.0D0
    SUM = ZERO
    TOL = ABS(TL)
    !---- SET INITIAL EIGENVECTORS -------------
    DO 200 I=1,NEQ
    DO 190 J=1,NEQ
    IF (TL.GT.ZERO) EIGVEC(I,J) = ZERO
    190 SUM = SUM + ABS(A(I,J))
    IF (TL.GT.ZERO) EIGVEC(I,I) = 1.0D0
    200 CONTINUE
    !---- CHECK FOR TRIVIAL PROBLEM ------------
    IF (NEQ.EQ.1) THEN
      EIGVAL(1) = 1.0d0
      RETURN
    END IF
    IF (SUM.LE.ZERO) RETURN
    SUM = SUM/(NEQ*NEQ)
    !-----------------------------------------
    !---- REDUCE MATRIX TO DIAGONAL ------------
    !-----------------------------------------
    400 SSUM = ZERO
    AMAX = ZERO
    DO 700 J=2,NEQ
    IH = J - 1
    DO 700 I=1,IH
    !---- CHECK IF A(I,J) IS TO BE REDUCED -----
    AA = ABS(A(I,J))
    IF (AA.GT.AMAX) AMAX = AA
```

```
SSUM = SSUM + AA
IF (AA.LT.0.1*AMAX) GO TO 700
!---- CALCULATE ROTATION ANGLE ----------
AA=ATAN2(2.0D0*A(I,J),A(I,I)-A(J,J))/2.0D0
SI = SIN(AA)
CO = COS(AA)
!---- MODIFY "I" AND "J" COLUMNS --------
DO 500 K=1,NEQ
TT = A(K,I)
A(K,I) = CO*TT + SI*A(K,J)
A(K,J) = -SI*TT + CO*A(K,J)
TT = EIGVEC(K,I)
EIGVEC(K,I) = CO*TT + SI*EIGVEC(K,J)
500 EIGVEC(K,J) = -SI*TT + CO*EIGVEC(K,J)
!---- MODIFY DIAGONAL TERMS -------------
A(I,I) =  CO*A(I,I) + SI*A(J,I)
A(J,J) = -SI*A(I,J) + CO*A(J,J)
A(I,J) =  ZERO
!---- MAKE "A" MATRIX SYMMETRICAL -------
DO 600 K=1,NEQ
  A(I,K) = A(K,I)
  A(J,K) = A(K,J)
600 CONTINUE
!---- A(I,J) MADE ZERO BY ROTATION ------
700 CONTINUE
!---- CHECK FOR CONVERGENCE -------------
IF(ABS(SSUM)/SUM .GT. TOL) GO TO 400

DO I=1,NEQ
  EIGVAL(I) = A(I,I)
END DO

RETURN
END SUBROUTINE
```

Appendix C: Computation of an integrand containing cot(θ/2) and ln(θ) singularities at θ = 0

By Colin Pickthall

The integral that is causing numerical problems when computing the integrand at θ = 0 is (see Equation 37 in Wilson [3]):

$$F_n(\eta) = \frac{16}{\pi}\int_0^{\eta}\left\{\cos(\theta)\cos([2n+1]\theta)\ln\left[\cot\left(\frac{\theta}{2}\right)\right]\sqrt{\cos^2(\theta)-\cos^2(\eta)}+\sin(\eta)\ln(\theta)\right\}d\theta + \frac{16}{\pi}\eta\sin(\eta)[1-\ln(\eta)]$$

This equation can be rewritten as:

$$F_n(\eta) = \frac{16}{\pi}I_n(\eta) + \frac{16}{\pi}\eta\sin(\eta)[1-\ln(\eta)]$$

$$I_n(\eta) = \int_0^{\eta}\left\{\cos(\theta)\cos([2n+1]\theta)\ln\left[\cot\left(\frac{\theta}{2}\right)\right]\sqrt{\cos^2(\theta)-\cos^2(\eta)}+\sin(\eta)\ln(\theta)\right\}d\theta$$

In the above equation, note that θ runs from 0 to η, thus $\theta \le \eta$ and therefore $\cos^2(\theta) \ge \cos^2(\eta)$.

The first term of the integrand has three factors, as follows:

$$I_n(\eta) = \int_0^{\eta}\{f_1(\theta,n)\times f_2(\theta)\times f_3(\theta,\eta)+\sin(\eta)\ln(\theta)\}d\theta$$

$$f_1(\theta,n) \equiv \cos(\theta)\cos([2n+1]\theta)$$

$$f_2(\theta) \equiv \ln\left[\cot\left(\frac{\theta}{2}\right)\right] = \ln\left[\frac{\cos(\theta/2)}{\sin(\theta/2)}\times\frac{2\cos(\theta/2)}{2\cos(\theta/2)}\right] = \ln\left[\frac{\cos(\theta)+1}{\sin(\theta)}\right]$$

$$f_3(\theta,\eta) \equiv \sqrt{\cos^2(\theta)-\cos^2(\eta)} = \sqrt{\cos^2(\theta)-1+1-\cos^2(\eta)} = \sqrt{\sin^2(\eta)-\sin^2(\theta)}$$

In the limit as θ→0, these factors have the following limits:

$$f_1(\theta,n) \xrightarrow[\theta\to 0]{} 1$$

$$f_2(\theta) \xrightarrow[\theta\to 0]{} \ln\left[\frac{2}{\theta}\right] = -\ln(\theta)+\ln(2)$$

$$f_3(\theta,\eta) \xrightarrow[\theta\to 0]{} \sin(\eta)$$

The product of these three factors becomes $-\sin(\eta)\ln(\theta)+\sin(\eta)\ln(2)$. The first part of this cancels with the second term of the integrand, and so, for vanishingly small θ, the integrand in $I_n(\eta)$ becomes simply the constant value $\sin(\eta)\ln(2)$.

It is instructive and useful to look at the small-θ series expansions of the three factors. The following series are used:

$$\sin(\theta) \xrightarrow[\theta\to 0]{} \theta - \frac{1}{6}\theta^3 + \frac{1}{120}\theta^5 - \cdots \quad \cos(\theta) \xrightarrow[\theta\to 0]{} 1 - \frac{1}{2}\theta^2 + \frac{1}{24}\theta^4 - \cdots$$

$$(1-\varepsilon)^{1/2} \xrightarrow[\varepsilon\to 0]{} 1 - \frac{1}{2}\varepsilon - \frac{1}{8}\varepsilon^2 - \cdots \quad \ln(1-\varepsilon) \xrightarrow[\varepsilon\to 0]{} -\varepsilon - \frac{1}{2}\varepsilon^2 - \frac{1}{3}\varepsilon^3 - \cdots$$

The small θ series expansions of the three factors above, on using these series expansions, become:

$$f_1(\theta, n) \xrightarrow[\theta\to 0]{} 1 - \left[2n^2 + 2n + 1\right]\theta^2 + \frac{1}{3}\left[2n^4 + 4n^3 + 6n^2 + 4n + 1\right]\theta^4 + \cdots$$

$$\cot\left(\frac{\theta}{2}\right) \xrightarrow[\theta\to 0]{} \frac{2}{\theta}\left[1 - \frac{1}{12}\theta^2 - \frac{1}{720}\theta^4 + \cdots\right]$$

$$f_2(\theta) \xrightarrow[\theta\to 0]{} \ln\left[\frac{2}{\theta}\right] + \ln\left[1 - \frac{1}{12}\theta^2 - \frac{1}{720}\theta^4 + \cdots\right] = -\ln(\theta) + \ln(2) - \frac{1}{12}\theta^2 - \frac{7}{1440}\theta^4 + \cdots$$

$$f_3(\theta, \eta) \xrightarrow[\theta\to 0]{} \sin(\eta)\left[1 - \frac{1}{2\sin^2(\eta)}\theta^2 + \left(\frac{1}{6\sin^2(\eta)} - \frac{1}{8\sin^4(\eta)}\right)\theta^4 + \cdots\right]$$

These series can now be multiplied together. Although terms of order θ^4 have been retained above, they have not been retained below because the coefficients become unwieldy.

$$\begin{aligned} f_1(\theta, n) \times f_2(\theta) \times f_3(\theta, \eta) = &-\sin(\eta)\ln(\theta) + \sin(\eta)\ln(2) \\ &+ \sin(\eta)\left(\frac{1}{2\sin^2(\eta)} + 2n^2 + 2n + 1\right)\theta^2\ln(\theta) \\ &- \sin(\eta)\left[\frac{1}{12} + \ln(2)\left(\frac{1}{2\sin^2(\eta)} + 2n^2 + 2n + 1\right)\right]\theta^2 + \cdots \end{aligned}$$

It is seen that the first term above cancels with the second term in the integrand of $I_n(\eta)$, which eliminates the subtraction of diverging terms in the small θ limit evaluations of the integrand for numerical integrations.

One way to overcome the problem of evaluating the diverging terms is to split the integral into two parts, as follows, with the second part causing no problems for numerical integration:

$$I_n(\eta) = \int_0^{\delta} f(\theta, n, \eta)\mathrm{d}\theta + \int_{\delta}^{\eta} f(\theta, n, \eta)\mathrm{d}\theta$$

$$f(\theta, n, \eta) \equiv f_1(\theta, n) \times f_2(\theta) \times f_3(\theta, \eta) + \sin(\eta)\ln(\theta)$$

The value of δ should be chosen small enough so that the power series above provides sufficient accuracy: terms of order θ^4 and $\theta^4\ln(\theta)$ should be negligible. In this case, the first integral can be written as:

$$\int_0^{\delta} f(\theta, n, \eta)\mathrm{d}\theta \approx \int_0^{\delta} \sin(\eta)\left\{\ln(2) + \alpha\theta^2 \ln(\theta) - \beta\theta^2\right\}\mathrm{d}\theta$$

$$\alpha = \frac{1}{2\sin^2(\eta)} + 2n^2 + 2n + 1$$

$$\beta = \frac{1}{12} + \ln(2)\left(\frac{1}{2\sin^2(\eta)} + 2n^2 + 2n + 1\right)$$

This integral can be evaluated analytically, resulting in:

$$\int_0^{\delta} f(\theta, n, \eta)\mathrm{d}\theta \approx \sin(\eta)\left[\delta\ln(2) + \frac{\alpha}{9}\delta^3\left(3\ln(\delta) - 1\right) - \frac{\beta}{3}\delta^3\right]$$

$$= \delta\sin(\eta)\left[\ln(2) + \frac{1}{9}\delta^2\left(3\alpha\ln(\delta) - \alpha - 3\beta\right)\right]$$

For small values of η, care needs to be exercised because the coefficients α and β then become large. However, as long as δ is chosen much smaller than η, any potential problems should be able to be avoided.

DEFENCE SCIENCE AND TECHNOLOGY GROUP
DOCUMENT CONTROL DATA

1. DLM/CAVEAT (OF DOCUMENT)

2. TITLE

Linear-Elastic 2D and 3D Finite Element Contact Analysis of a Hole Containing a Circular Insert in a Fatigue Test Coupon

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Witold Waldman

5. CORPORATE AUTHOR

Defence Science and Technology Group
506 Lorimer St
Fishermans Bend Victoria 3207 Australia

6a. DST NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DST-Group-TR-3134	AR-016-357	Technical Report	July 2015

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
2015/1030263/1	AIR 07/283	OIC-ASI-DGTA	90	28

13. OBJECTIVE ID	14. RELEASE AUTHORITY
AV9143353	Chief, Aerospace Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

Yes

18. DST GROUP RESEARCH LIBRARY THESAURUS

Contact analysis, Stress concentration, Holes, Fatigue life, Numerical modelling, Numerical simulation, Finite element analysis, Aircraft structure

19. ABSTRACT

Aircraft structures typically contain large numbers of circular holes that are fitted with fasteners such as bolts or rivets. During the service life of aircraft, fatigue damage often occurs at such holes. The accurate analysis of stress distributions occurring around the boundary of holes in the presence of fasteners is therefore an important consideration during studies of fatigue life and test interpretation activities supporting full-scale fatigue test programs. In the present work, two-dimensional linear-elastic plane elasticity solutions for contact stresses caused by a circular disk inserted into a circular hole in an infinite plate undergoing remote loading have been implemented in a FORTRAN program. These were used to validate the contact stress distributions for a circular hole in an aluminium plate fitted with a titanium fastener that were computed using two-dimensional finite element contact analysis. By application of a finite-width correction factor, the analytical infinite-plate solutions were also used as a point of comparison with the results produced by subsequent two-dimensional and three-dimensional finite element contact analyses of a finite-width fatigue test coupon. The results obtained here are useful for aircraft structural integrity analysis work, and subsequent analyses of contact problems such as this one can be expected to be accurate so long as sufficiently refined finite element meshes are utilised.