Mathematical Modelling of Instruments for Pressure Metrology

Noel D. Samaan

A thesis submitted for the degree of
Doctor of Philosophy

City University
Measurement and Instrumentation Centre
Department of Electrical, Electronic and Information Engineering
School of Engineering
Northampton Square
London EC1V 0HB

June 1990

to

my son Fady

my wife Mona

our families

and my friend Faruq

CONTENTS

ABSTRACT

The development and application of mathematical models for instruments used in pressure metrology are presented. Applications to industrial instrument design are followed by the use of finite element analysis applied to the modelling of an optically driven vibratory sensor. Results for the latter compare favourably with measurements on an experimental prototype.

For pressure standards metrology, an interactive program, **PVE** (**P**ressure **V**iscosity **E**lasticity), has been developed to simulate oil and gas operated pressure balances, inclusive of both pressure-elasticity and pressure-viscosity effects. The program uses "unit" load data generated by a purpose-built **FEA** (**F**inite **E**lement **A**nalysis) program to characterise a given pressure balance. Then by iteration, the pressure and the piston-cylinder gap profiles along the engagement length are computed and displayed graphically. The distortion coefficient, λ, defined by the effective area equation $A = A_0(1 + \lambda P)$, is then calculated from the piston and cylinder elastic distortions and the pressure profile.

The PVE program has been applied to the NPL range of (simple geometry) pressure balances (series 100, 200, 300 and 400). Results show that λ is essentially constant for each of the balances. For the series 100 and 200 balances, λ is approximately 3.3 ppm/MPa whereas for the series 300 and 400 balances its value is 3.0 ppm/MPa. The oil viscosity variations along the engagement length are large for the series 400 balance operating at its full pressure (320 MPa) but even in this case, λ remains constant. Applications were also made to two other designs, a RUSKA 2481 re-entrant oil operated balance and a RUSKA 2470 re-entrant gas operated balance. For the oil operated one, λ was found not to be constant and the effective area, A, needed to be approximated with a cubic polynomial. Consequently, for this balance, λ varied between -0.7 and -1.94 ppm/MPa in the range 28 to 280 MPa. For the gas operated balance, λ was found to be constant at 0.14 ppm/MPa in its operating range up to 17 MPa.

The PVE and FEA programs are both implemented on a graphics workstation. The theoretical approach adopted and the programs developed are discussed and presented.

Acknowledgements

The author gratefully acknowledges both of his supervisors, Dr. F. Abdullah and Dr. K.T.V. Grattan for constant guidance and help throughout this Ph.D. study. Dr. Grattan contributed through his guidance and useful comments in the optical sensor work and Dr. F. Abdullah contributed to all aspects of the mathematical modelling, the primary concern of this work.

The modelling of pressure balances has been supported, in its early phase, by an Extra Mural Research Agreement (EMRA) with the U.K. National Physical Laboratory (NPL). This is gratefully acknowledged alongside useful discussions with Dr. Peter Stuart, Mr. David Simpson and Mr. Michael White of the pressure and vacuum section at NPL. Also, acknowledgements are to the Scientific Research Council, Baghdad, Iraq for financial support.

Throughout this work, and especially in the final stages, I much appreciate the help of my colleagues, Dr. M.K. Mirza and Mr. Z. Mouaziz and the departmental secretaries Mrs. Linda Carr and Mrs. Joan Rivellini.

Finally, to Dr. Faruq Abdullah, I would like to pay a special tribute for his inspiration, patience and willingness to discuss, *at any time*, problems I may have had, both technical and others. In short, it has been a great joy for me to have worked with him.

List of Tables

List of figures

CHAPTER 1

Introduction

Pressure is one of the most important variables to be measured in scientific and industrial applications. It is a quantity which cannot be measured directly but is derived from base units through definitions (i.e force acting upon an area). The unit of pressure is the Pascal (Pa). In general, pressure measuring instruments can be divided into two main groups, those that are direct and indirect. The first group determines the value of applied pressure by directly calculating the force applied upon an accurately known area whereas instruments of the second group are based on transforming pressure to some other variable which is then measured (e.g. the use of elastic elements to produce displacements or strains). The latter instruments are always calibrated using the first group such as manometers or dead weight testers, also known as *pressure balances*. The pressure balance in particular has found wide application in pressure metrology and forms the primary standard for measuring pressure over a range of four orders of magnitude from 0.1 MPa (1 bar) to 1000 MPa. Only in the region 0.1 MPa is the pressure balance surpassed in its accuracy by the mercury manometer (figure 1.1).

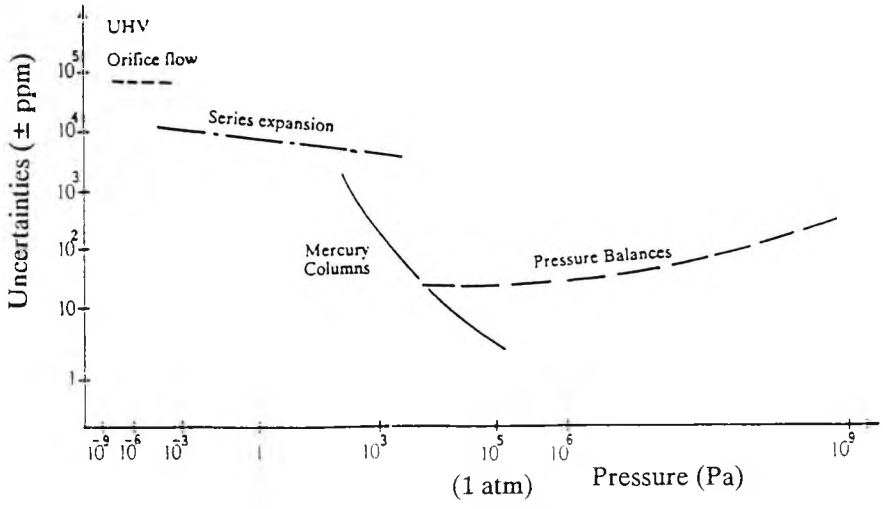


Figure 1.1. Uncertainties in NPL pressure standards.
(Taken from NPL booklet on "pressure and vacuum", National Physical Laboratory, Teddington, U.K., 1984).

When used for calibration, a pressure balance may be considered as a pressure generating instrument which is used to calibrate pressure measuring instruments such as those based on the deformation of an elastic sensing element. The latter form the bulk of industrial instruments used for non-vacuum pressure measurement. Typical highest accuracies for industrial instruments are now 0.1 % (1000 ppm) uncertainties and directly reflect the improvements in primary standards over last few decades. Typical errors for pressure balances are now such that they show less than 250 ppm uncertainty over the entire range from 0.1 to 1000 MPa. The primary reason for this high accuracy for the pressure balance is that distortion effects, which become the dominant error at higher pressures, can be corrected. These distortions change the piston and cylinder geometries, so the effective area, A, arising from the balance of forces on a loaded piston, given by:

$$P = \frac{m \cdot g}{A}$$

is *not* constant but depends on a linear term in pressure P. So,

$$A = A_o (1 + \lambda P) \qquad (1.1)$$

where A_o is the nominal area and λ is known as the distortion coefficient (reference [1]).

Since λ can be of the order of 3×10^{-6} (3 ppm) per MPa, the need to determine λ accurately particularly for high pressure balances is apparent. The confidence to which λ is known experimentally is very variable and uncertain as emerged from a recent international comparison of high pressure balances. This was pointed out at a recent meeting of the BIPM high pressure group in Paris (Molinar et al [2]). Although theoretical work for determining λ has been considered previously, it has been only of

limited generality in being incapable of handling balances of complex geometries. However, numerical methods can be applied to more general cases.

Consequently, this thesis is concerned with the development of a theoretical numerical approach for the mathematical modelling of pressure balances to determine the distortion coefficient λ. The use of numerical modelling techniques within engineering, applied to industrial design, is now widespread. Within instrumentation, mathematical models are finding application for sensor and transducer design.

Chapter 2 reviews this area highlighting the success of the Computer Aided Design of Instruments (**CADI**) research group at the Measurement and Instrumentation Centre of the City University. This includes design methods for conventional pressure sensors with industrial applications. The chapter also introduces the Finite Element (**FE**) method commonly used in CAD design of instruments. The FE method and other analytical techniques were applied to simulate optically driven resonant sensors and results compared with experiment.

In chapter 3 the pressure balance is introduced, previous work is reviewed and the theoretical approach adopted to the mathematical modelling is developed. This includes the translation of the approach into the formulation of the **PVE** (**P**ressure-**V**iscosity- **E**lasticity) program which uses data generated by a purpose-built **F**inite **E**lement **A**nalysis (**FEA**) program. Both programs form the numerical model which allows the calculation of the distortion coefficient incorporating both elastic and fluid pressure- viscosity effects.

A primary purpose of this thesis was to show the adequacy (or otherwise) of equation 1.1 which represents the effective area with **only** a linear term in pressure as well as calculating the actual value of λ for a number of primary balances. Chapter 4 applies the model to the U.K. National Physical Laboratory (NPL) series of primary oil operated pressure balances and two RUSKA designs (of RUSKA Instruments Corporation /U.S.A.): one oil operated and one gas operated. Extensive results for all these balances are presented in this chapter including graphical display of quantities of interest.

Finally, chapter 5 summarises the work and points out the potential contribution to the future understanding and design of pressure balances.

CHAPTER 2

Design of Pressure Sensors using Mathematical Models

2.1 Introduction

The mathematical modelling of pressure balances to be described in the next chapter builds on the expertise developed within the Computer Aided Design of Instruments (CADI) research group at City University. In this chapter the area of modelling and design of industrial pressure sensors using diaphragms and capsules is reviewed. This was an early area of success for the CADI group which highlighted the conceptual design breakthroughs that can be made once appropriate mathematical models have been developed. The modelling of diaphragms and capsules involved the application of finite difference methods to solve the appropriate differential equations of thin shell theory and the subsequent development of design methods based on the modelling. More recent mathematical modelling has extended the field beyond pressure sensing using principally the finite element (FE) method (reviewed in [3,4]). Since the FE method is adopted for modelling of pressure balances (as explained in the next chapter), section 2.3 of this chapter introduces the FE method including modern trends in its use on interactive graphics workstations.

As part of this work, techniques of mathematical modelling, including finite element analysis, were applied to optically driven miniature resonating structures. With optical drive through optical fibres, a potential application is to low cost and intrinsically safe

pressure sensors. The modelling is described briefly in section 2.4. Fuller details are provided in a paper published by the author and appended to this thesis.

2.2 Diaphragms and Capsules

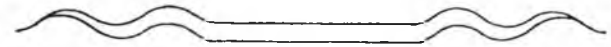
Diaphragms and capsules are widely used as primary sensors within industrial pressure measuring instruments. Modern instruments of this type have low errors (typically better than 0.5% uncertainty) and are of complex construction often involving the use of corrugated diaphragms or capsules made of two corrugated diaphragms edge welded at their periphery (figure 2.1). The corrugations are introduced to ensure a linear pressure deflection characteristic over the required pressure range (typically 1 bar). Thus the centre deflection is an accurate representation of the applied pressure. Diaphragms are characterised by variables such as thickness, overall diameter, number and height of corrugations, Young's modulus, Poisson's ratio etc. The linearity of a given diaphragm (or capsule) is very difficult to predict since no adequate analytical theory exist for these devices. Consequently, in the past, the design of pressure sensors based on these elastic elements was costly and time consuming. For example, the design of nesting capsules could take up to a year because of the three main requirements: (i) given sensitivity, (ii) given non- linearity error and (iii) "nesting" of the top diaphragm onto the bottom diaphragm under overload pressures. The process of design would require several iterations to achieve these requirements. Each iteration would require tooling, manufacture and testing of capsules.

Mathematical models for diaphragms and capsules were developed in an early thesis (Turley [5]) and shown capable of predicting sensitivity and linearity. These models were based on the solution, by finite difference methods, of Reissner's differential

equations for a thin shell. The work illustrated the nature of the use of mathematical models for gaining insight into diaphragm and capsule performance with the subsequent emergence of a methodology for the design of nesting capsules. Before a mathematical model for a sensor (pressure or otherwise) can be used for design it is necessary to specify those variables that a designer is free to vary within a model. This requires geometric characterisation. For example for a corrugated diaphragm, design variables would be diaphragm thickness, diameter, number and height of corrugations and these must be incorporated into a mathematical model. Sensitivity analysis, by computer simulation, would then be used to determine which of these variables are important and which can be neglected for the purpose of design. In this way the mathematical models (which had been validated by experiment at an earlier phase) can be used efficiently and the design process can be automated. Further, having appropriate mathematical models can open up possibilities for better design. This is illustrated for the design of perfectly nesting capsules in what some designers have previously called a "black art". Since the computer model developed in reference [5] allows the deformation of the diaphragms (forming the capsule) to be traced, the insight is to *start* with capsule nested so forming a "diaphragm" (figure 2.1d).

This nested capsule is "blown apart" through computer simulation and the top and bottom diaphragms are then "frozen". These form the capsule which is then tested under compression and extension for linearity and sensitivity over the desired pressure range. If unsatisfactory, the parameters of the nested "diaphragms" are altered systematically until a satisfactory design is achieved. With this method, perfect nesting is guaranteed under overload. In fact, in the early industrial designs, nesting was so good that in liquid filled capsule-stacks used in differential pressure measurement, a time lag of minutes occurred before the diaphragms in a capsule stack would release (due to the viscosity of the liquid). In later designs the end convolutions were modified

a



c

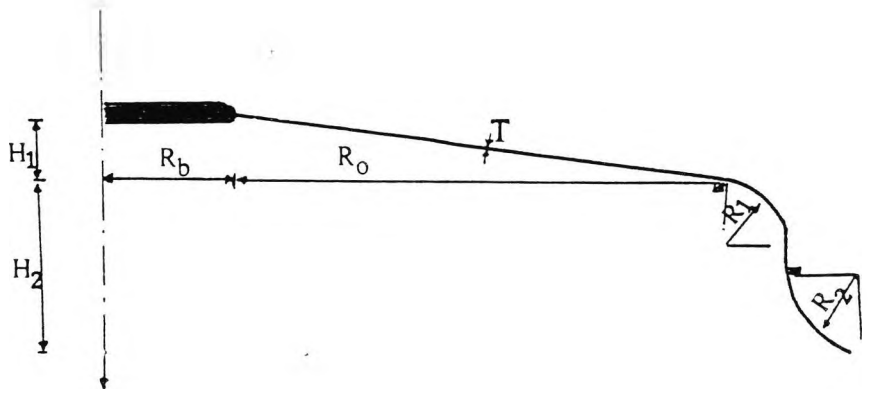
b

d

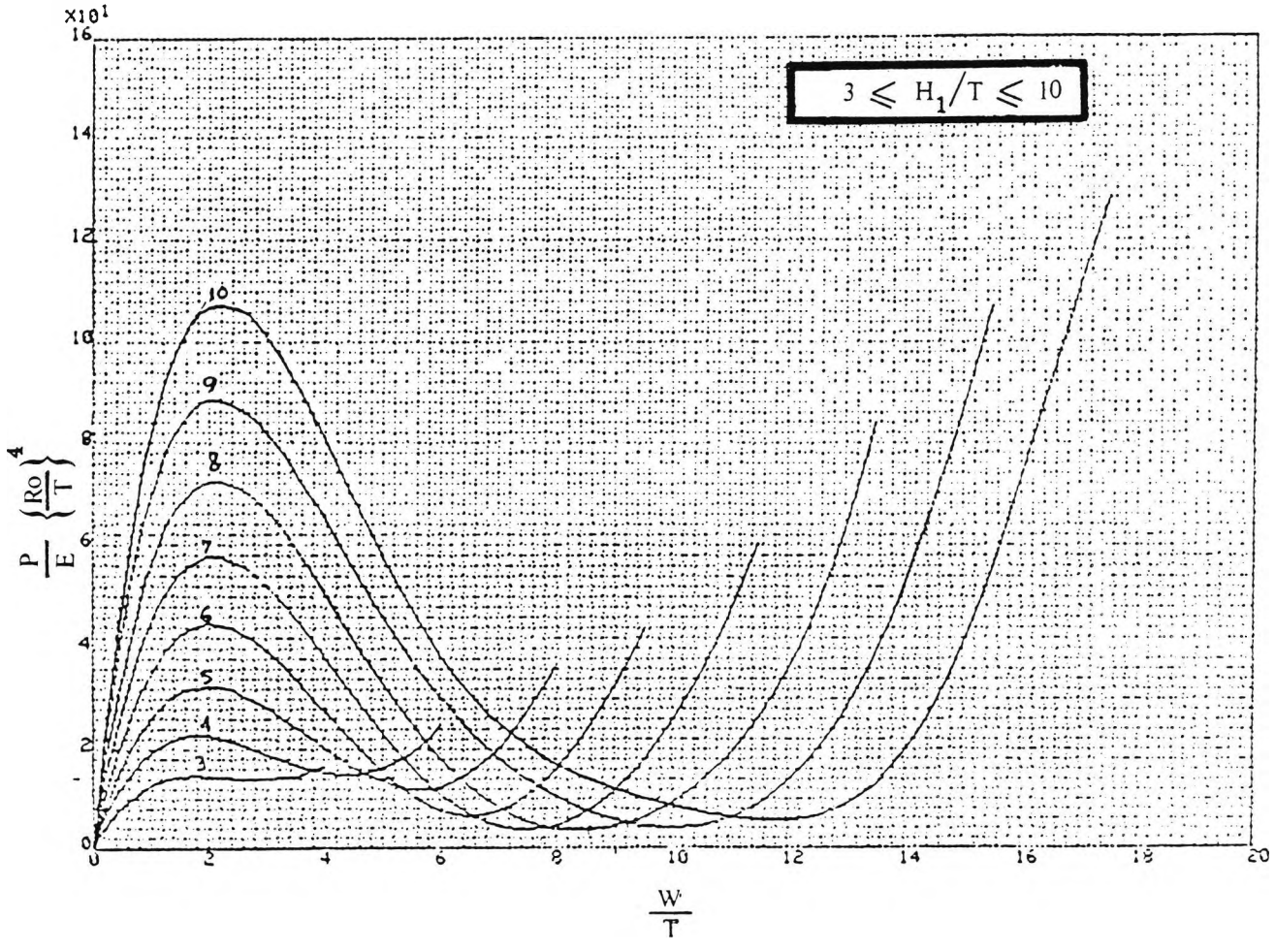
Figure 2.1. CAD of pressure capsules: (a) designed capsule, (b) under rated extension, (c) under rated compression, (d) under 250% compressive overload showing perfect nesting (ref. [4]).

and the simulation repeated. A design of a single capsule is shown in figure 2.1, with the designed capsule, its shape under extension and compression load and its perfect nesting behaviour under a designed 250% overload in pressure.

Closely related pressure sensing elements are snap action diaphragms. A snap action diaphragm is basically a thin shell diaphragm experiencing an in-plane tensile stress due to an externally applied pressure. A schematic diagram (axisymmetric view) of a conical snap action diaphragm is shown in figure 2.2a, with the principal geometrical dimensions indicated. As the in- plane stresses increase under loading, the stiffness of the diaphragm increases. These diaphragms are generally used as force or pressure operated switches, ideally suited for these applications due to their positive action, large displacements and good reproducibility. An ordinary diaphragm shows a characteristic of progressive pressure against centre deflection. This means that as the applied pressure increases, the rate of displacement of the diaphragm centre is reduced. For the snap action diaphragm, a sudden change in the diaphragm profile, as pressure is increased, occurs with a loss of stability due to the increase of the compressive stresses. "Snap through" occurs at one pressure and "snap back" at a lower pressure resulting in hysteresis. To simulate these conditions, numerical methods (e.g. the Finite Element (FE) method) were most appropriate to use due to the complexity of the geometry, very large displacements and more importantly, the instability of the structure in order to predict the pressure-deflection characteristics. Obtaining the latter would be clearly an impossibility with the use of analytical methods. The model developed using the FE method was employed to generate dimensionless performance curves for such diaphragms which can lead to a simple design methodology. These performance curves are shown in figure 2.2b interrelating design parameters of the diaphragms [6].



a



b

Figure 2.2. Conical snap action diaphragms.
(a) geometrical characterisation, (b) computer generated non-dimensional performance curves (ref.[4,6]).
{E *is the Young's modulus and* W *is the centre deflection of the diaphragm*}.

Looking at these normalised performance curves, it can be seen that sensitivity analysis performed with the FE model led to the elimination of a large number of design variables {from eight to four (t, H1, Ro, E)}. The design methodology developed in reference [6] used these curves within a simple interactive program on a personal computer.

To summarise, in the development of models and their subsequent use for design, the following stages can be seen:-

1. Model development and its validation with existing (or constructed) sensors.

2. Geometrical characterisation of the sensor with a number of design variables.

3. Sensitivity analysis to determine the principal design variables (geometric and material).

4. The use of dimensional analysis to form dimensionless groups.

5. The production of performance curves.

6. The development of a methodology for design.

In some cases all the stages can be followed (e.g. for snap action diaphragms) whereas for others , this is not possible. For example for capsules, dimensionless groups cannot be formed due to geometric complexity (therefore, stages 4 and 5 are omitted).

A more detailed review of the work reported in this section can be found in reference [4] where interactive finite element modelling of load cells is also described.

2.3 The Finite Element (FE) Method and its use

The Finite Element (FE) method is a numerical method for solving field problems. A field problem can be structural (e.g. solving for displacements), thermal (e.g. solving for temperature) or fluid (e.g. solving for velocity). In a structural problem, for example, a structure is divided into a "finite" number of small units known as "elements".

This requires an idealisation of the problem to be considered and depending on this idealisation, the FE model can be represented either in one, two or three dimensions. Appropriate elements in the FE theory are available to represent the structure conveniently. For example, a beam element for a one dimensional representation, triangular or quadrilateral elements for a two dimensional representation and finally perhaps brick elements for representing the structure in three dimensions. In general, elements possess nodes on their boundaries and in some cases inside them. Once an element type is chosen for an idealised structure, the structure is represented by a number of elements and nodes associated with them. The elements are joined together via their corresponding boundary nodes (both assigned by integer number) to form the *mesh* which divides the structure. This is known as the discretisation of the structure. The unknown field variables (depending on the type of the field problem) are defined at nodes and the number of these variables at a node is known as the **d**egree **o**f **f**reedom (**dof**) per node (e.g. for a 2D structure, dof per node = 2, representing the displacements in the x and y directions at each node).

The number of dof in the system (the whole structure) is the product of the number of nodes and the number of dof per node. Similarly, the number of degrees of freedom

per element is the product of the number of nodes per element and the dof per node. Data are supplied to define the coordinates of each nodal point and the element topology (list of global node numbers associated with a particular element which in turn will define the mesh). Added to these data are: defining which of the nodal parameters at specific node(s) have boundary constraints and finally the material properties of the structure (Young's modulus, Poisson's ratio).

Having defined the structure in terms of elements and their associated nodes, the next step would be to specify the behaviour of each individual element, i.e. the displacement variations within the element. This is defined as a prescribed function in terms of the displacements (and possibly their derivatives) at the nodal points.

Thus, if the displacement at each node is known, the displacement anywhere within the element can also be extracted. Functions describing these displacements and their variations are *known* for each type of element and are called the *interpolation functions*. It is important to note that an element will behave in exactly the same way the displacement variations are *assumed*. In this way, the finite element (FE) solution will depend entirely on this idealisation, and consequently, the solution will only be as good as the idealisation itself. Following this, the next stage relates the variables of the nodal displacement to the corresponding forces. This is known as forming the *stiffness matrix of an element*; the more stiff, the more load is required to get the same displacement. Once the element behaviour is known, the behaviour of the entire structure can be defined in terms of the behaviour of individual elements. This is achieved by tying all elements together and forming the *system stiffness matrix*. It represents the contribution of each element in the structure. This will result a set of system equations which can be solved (after applying the necessary boundary conditions) for the unknown nodal variables (e.g. displacement) which in turn (in most cases) are required

to be displayed. In some problems, this stage would be sufficient to terminate the procedure while in other problems, depending on the requirement, the calculated values are used to determine other quantities of interest (e.g. stress, strain, etc).

In summary, modelling of any field problem using the FE method will involve the following steps:

1. Idealisation of the problem which will include simplifying the structure and deciding whether it can be represented in one, two or three dimensions. This will in turn lead to the selection of an appropriate type of element and an idealised model geometric dimensions, together with material properties.

2. Discretisation which involves generating an element mesh for the structure.

3. Boundary constraints which arise from support conditions and external loading (e.g. force, pressure etc).

4. Formulation of the appropriate equations representing the behaviour of the structure under consideration.

5. Solution of appropriate equations to obtain field variables at nodal points.

6. Further calculations to determine other quantities of interest from nodal variables. e.g. strain, stress etc.

7. Presentation of the results either numerically and/or graphically.

In modern terminology, steps 1 to 3 are known as the **pre-processing** stage, steps 4 and 5 as the **analysis** stage and finally steps 6 to 7 are known as the **post-processing** stage. In practice it is very likely that a field problem demands a considerable effort to

implement the three stages. Since the FE method is a numerical method and, by nature, it posses systematic and repetitive procedures, efficient methods have been developed for producing FE packages.

The recent availability of the graphics workstations with the power of (previous generation) super computers, including interactive graphic facilities, has geared the thrust in the development of FE packages to integration of all the three stages (preprocessing, analysis and post processing) in a highly interactive fashion. A package PEARS (Plane Elastic Analysis and Reaction Stresses) developed at Rutherford Appleton Laboratory (RAL) is an example of such an FE package. A structure can be entered graphically including the definition of boundary and loading conditions, followed by the solution of the system equations and finally a graphical presentation of the results, all with the use of "pop-up" menus. As an example, figure 2.3 shows the mesh of a typical cylinder of a pressure balance of simple geometry (explained in detail in chapter 3) together with the boundary (support) conditions marked as a series of x's and the pressure loadings marked by arrows, while figure 2.4 shows the distorted shape of the same geometry due to the application of a linear pressure along the innermost bore of the cylinder. It can be seen from figure 2.4 that, for example, the radial displacement can also be displayed in a graphical form on a separate sub-window using the pop-up menus. With such a package, the user requires very little knowledge about the FE method and a typical problem can be entered, solved and viewed in 30 minutes on a Sun/SPARC workstation. Of this time, the analysis stage takes only a few minutes. Therefore further explorations such as change of support conditions (boundary values) and/or loading can be done interactively in a single session.

On the other hand, a number of commercial FE package are available. These packages are integrated packages in the sense that they incorporate the preprocessing, analysis

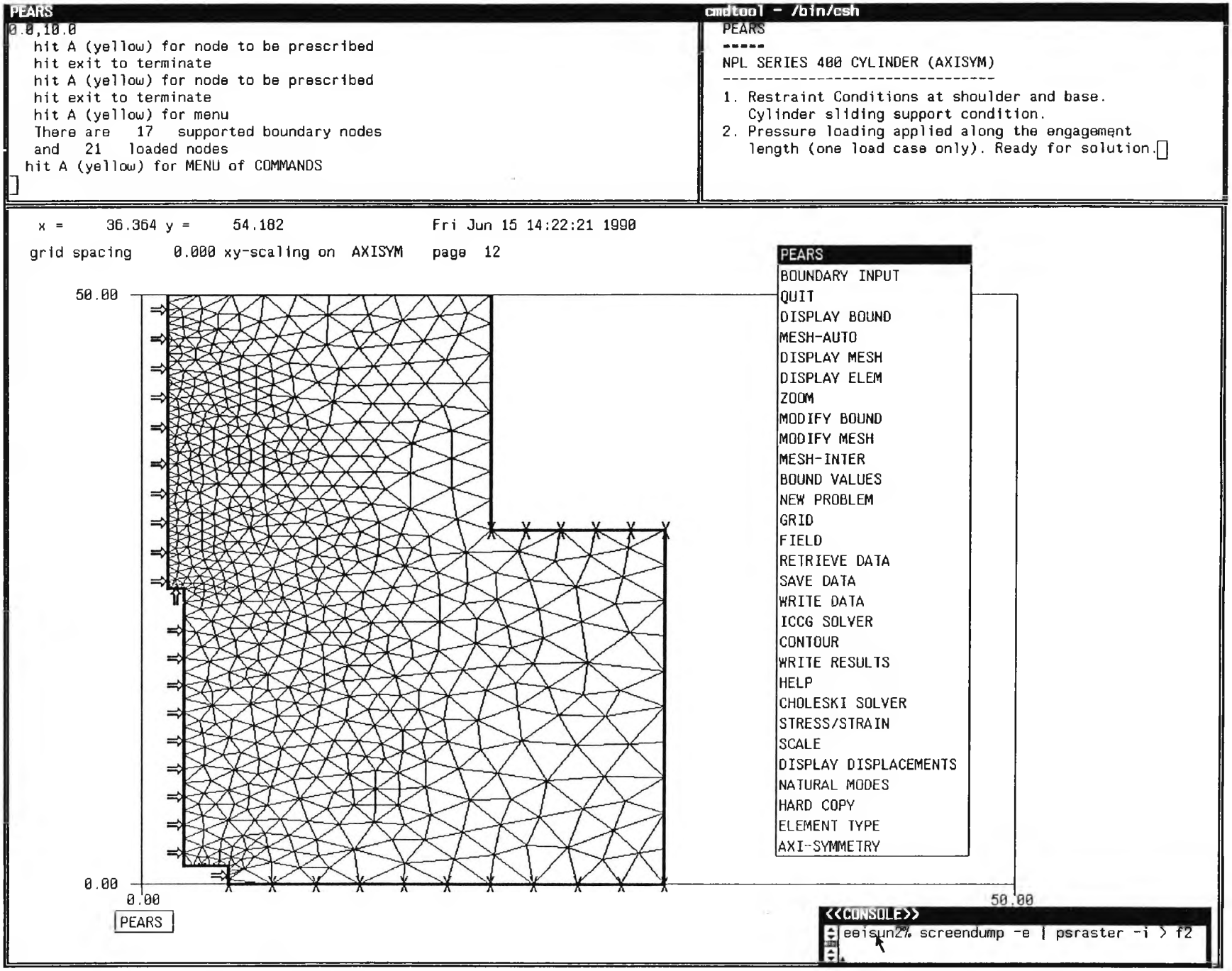


Figure 2.3. Element mesh for a pressure balance cylinder using an interactive FE package PEARS.

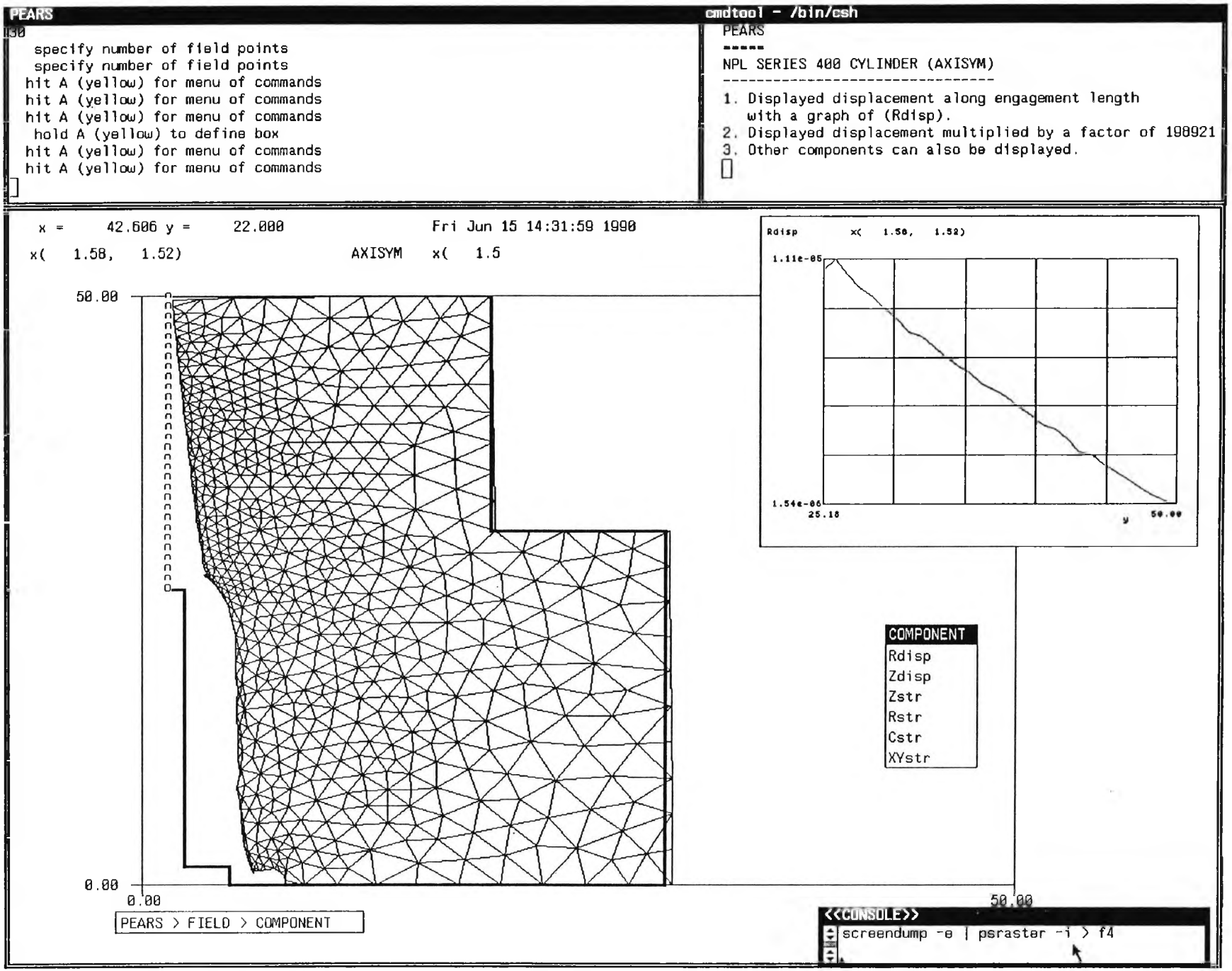


Figure 2.4. The elastic distortions and the radial displacement for a pressure balance cylinder,using PEARS.

and the post-processing stages. However, they are non-interactive packages and the user requires some knowledge about the FE method. In general, in order to use these packages, the geometry, material properties, boundary constraints and loading conditions are entered in an input file with a restricted format. This data file is "launched" for mesh generation, system equation set-up and solution of the nodal variable (and other quantities of interest) in an off-line process. The results are usually provided in an output file (in numerical form). Some of these packages have complementary viewing facilities which have to be "launched" at a later stage of the analysis stage. In terms of providing the user with a fairly large number of element types and accommodating the modelling of different field problems (structural, thermal, flow, etc), these packages are more flexible than the highly interactive integrated packages described earlier. However, the user has to keep track of (with some knowledge about the FE method) the node and element numbering, the way the boundary values and the loading conditions are interpreted and finally the difficulty which arises from having the post-processing stage (if available) as an off-line process. This is a disadvantage when comparing these packages with the interactive ones. An example of the non-interactive integrated packages is a package known as LUSAS which was used in the modelling of optically driven resonant structures explained in section 2.4. For the latter modelling procedure, it required a few hours to set the input file and to get the solution. Further, explorations could also take a considerable amount of time, and in fact, some of the analysis carried out (the photothermal analysis) took several hours.

As mentioned earlier, the mathematical modelling using the FE method, can be carried out using interactive (e.g. PEARS) or non-interactive (e.g. LUSAS) packages. Although these types of packages, when combined, provide a great flexibility in modelling of field problem, yet they are often limited in meeting the full requirement

of the modelling exercise. This is evident in the sense that, by definition, the more general a package is, the less specific it becomes since the analysis stage is not accessible to the user. This situation leads to the requirement of developing purpose built FE analysis programs around which application specific pre- and post-processing programs can be developed. These type of FE analysis programs are increasingly based on standard library sub-routines. For example, the purpose built FE Analysis (FEA) program developed for the modelling of pressure balances (as explained in Chapter 3) was based in the NAG/FE library.

The finite element library (FELIB), known as the NAG/SERC Finite Element library, was originally developed at Rutherford Appleton Laboratory (RAL) with the aim of providing a tool for the development of purpose built FE analysis programs in a relatively simple and efficient manner with reasonable flexibility. The FELIB is based on FORTRAN and is supplied in two parts known as LEVEL 0 and LEVEL 1.

LEVEL 0 is a set of subroutines tailored specifically for the use in FE analysis programs. They facilitate the modeller with a tool to construct systematically the structure of an FE analysis program such as choosing the appropriate element type, its associated interpolation functions, determine the stiffness matrix of that element, assemble (tie together) all elements of a geometry resulting in the system stiffness matrix and other complementary routines such as integration routines, matrix manipulation (addition, multiplication, inversion etc) and system matrix solution. Also, other routines are available to handle a standard form for reading-in the geometry mesh (coordinates and topology), loading data and for writing (outputting) the calculated parameters (such as displacement, stress, strain etc) to a device or a file. Details of LEVEL 0 subroutines are documented in reference [7].

LEVEL 1 library consists of example programs for solving static and dynamic problems [7]. These example programs use routines available in LEVEL 0 library and they (example programs) attempt to bridge the gap between theory and practice (i.e. programming techniques) of the FE method. The programs are portable and have a modular structure so that modifications are made with little difficulty to suit the requirements of the problem. In this way, the overall structure of an FE analysis program can be defined and the time taken (from problem definition to producing a working analysis program) can be reduced if the modeller selects the appropriate (example) program that is close to his/her problem.

2.4 Modelling of Optically Driven Resonant Sensors

The use of resonating structures in optical sensing systems has been proved to be one of the successful approaches to the measurement of physical parameters [8]. These sub-systems are often based on relating the measurand to the change in the resonant frequency of a structure which is set into a flexural mode of vibration. With optically-driven resonators, the optical power is modulated at a frequency equal to the resonant frequency of the structure. Light is converted into thermal energy due to the photothermal conversion which takes place at the surface of the vibrating structure since the latter is coated with an absorbent material, usually evaporated onto the structure. Recently these techniques are being applied to silicon microstructures for pressure sensing (see Uttamchandani et al [9]).

Although a considerable amount of experimental work in the field has been reported by a number of groups, as yet comparatively, little theoretical modelling work is done. The background expertise in mathematical modelling of sensors including finite

element techniques (explained earlier) was applied to this area. Details are provided in a publication (Grattan et al [10]) which is appended to this thesis. A particular device previously investigated (see references [11,12]) has been chosen for modelling.

The schematic diagram of the mounted vibrating (quartz crystal) structure is shown in figure 2.5, while figure 2.6 shows a cross section through the thickness of the (quartz) structure. As indicated, a coating of an absorbent material covers one side of the structure, suitably chosen to absorb the maximum light energy at a particular wavelength (820 nm, near infrared). The overall size of the quartz structure was 12.5 mm long, 6 mm wide and 125 μm thick.

The objectives of the work consisted of :

a. The development of a general purpose computer program to simulate the photothermal conversion to study the sensitivity of the process and the temperature profile through the thickness of the vibrating structure, both analytically and numerically.

b. A theoretical determination of the maximum deflection of the structure, its resonant frequency, mode shapes, prediction of the frequency/load relationship and the effect of the size of the optical fibre on the maximum deflection.

The photothermal analysis required solving Fourier heat equations governing the photothermal conversion, then the a.c. temperature rise at the surface of the vibrating structure can be calculated as well as the temperature profile through its thickness which in turn will govern the deflection of the structure. A typical temperature profile is shown in figure 2.7.

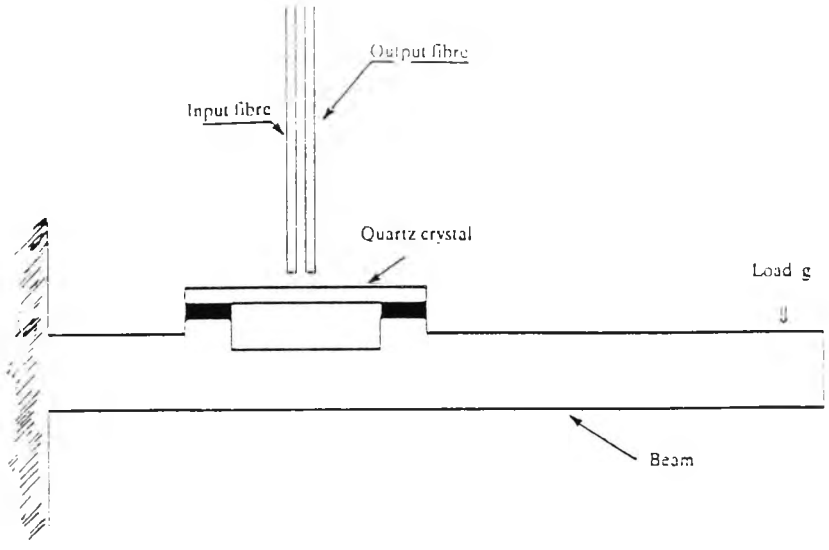


Figure 2.5. Schematic diagram of a vibrating quartz element mounted on a beam indicating the excitation and sensing fibres.

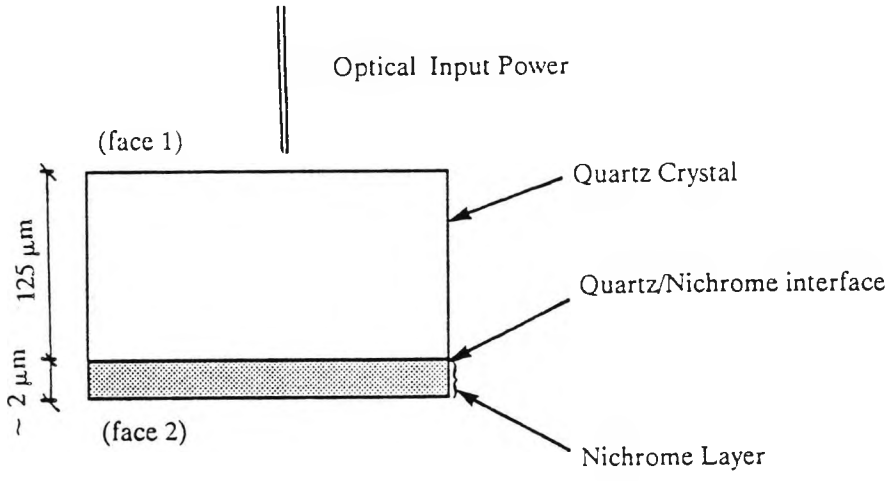


Figure 2.6. A section through the thickness of the vibrating quartz element.

The photothermal model was extended (subsequent to publication of reference [10]) to implementation of other possible schemes for driving the structure, namely, applying modulated optical power directly onto the coated surface rather than via the non-coated one. The model also accommodated absorption in the vibrating structure, if any.

Further, the photothermal model also allowed the varying of a number of input parameters interactively with graphical presentation of the temperature profile during one cycle of the input optical signal.

On the other hand, the structural analysis included the use of analytical and numerical methods to determine the frequency of vibration, the deflection, the mode shapes and prediction of the frequency/load relationship.

Static deflection was then determined by three methods:

i. by approximating the structure to a simple beam with a linear temperature gradient across its thickness,

ii. by treating the structure as a bimetallic strip and

iii. using the Finite Element (FE) method with shell elements provided with a commercial package LUSAS [13].

Once the static deflection is determined using any one of the above methods, and as the system is experiencing a flexural mode of vibration, it can be approximated to a

single degree of freedom under forced vibration. The dynamic deflection could then be calculated by multiplying the static deflection by the Q factor of the system.

Applying the FE method, the resonant frequency of the structure, its mode shapes of vibration and the frequency/load relationship were determined.

In summary, for an average input optical power of 4mW (at a wavelength of 820 nm) and Q factor of 2700, the dynamic deflection was found to be 0.65 μm, 0.59 μm and 0.27 μm in the case of the simple beam, bimetallic strip and the FE method respectively. This result is to be compared with an experimental result of between 1 μm and 2 μm, and a theoretical value of 50 μm reported by Mallalieu [12] (for more details see reference [10]). On the resonant frequency, the calculation using the FE method agreed within 23% of the experimental results while the discrepancy in the slope of the frequency/load relationship was 8% but linearity was confirmed (see figure 2.8). These discrepancies can be due to errors in dimensions and composition of the coated material resulting from an evaporation process.

The models developed for the modelling of the quartz structure have been applied to a typical Si micromachined diaphragm structure (760μm x 760μm x 4μm), used as a pressure sensor [9]. In particular, the effect of the size of the fibre carrying the modulated optical power was investigated. This was done by varying the area on which the (optically produced) thermal load is distributed on the diaphragm and calculating the corresponding deflection (using an FE model of the diaphragm). It was found that a significant change in deflection can occur if a 500 μm fibre was used instead of a 100 μm. In this case, the deflection can drop by about 60% if the larger fibre was used for a fixed geometry and average input power.

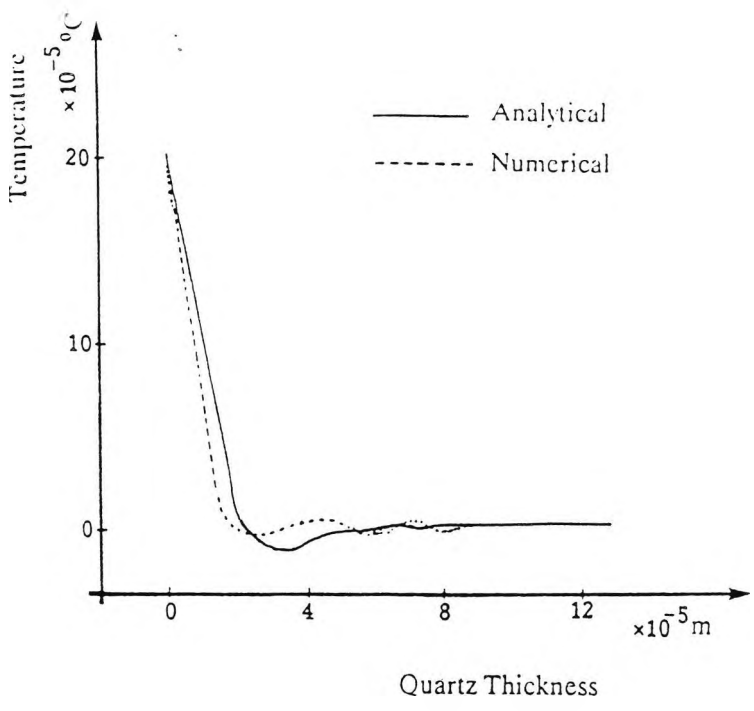


Figure 2.7. A typical temperature profile through the thickness of a vibrating quartz element.

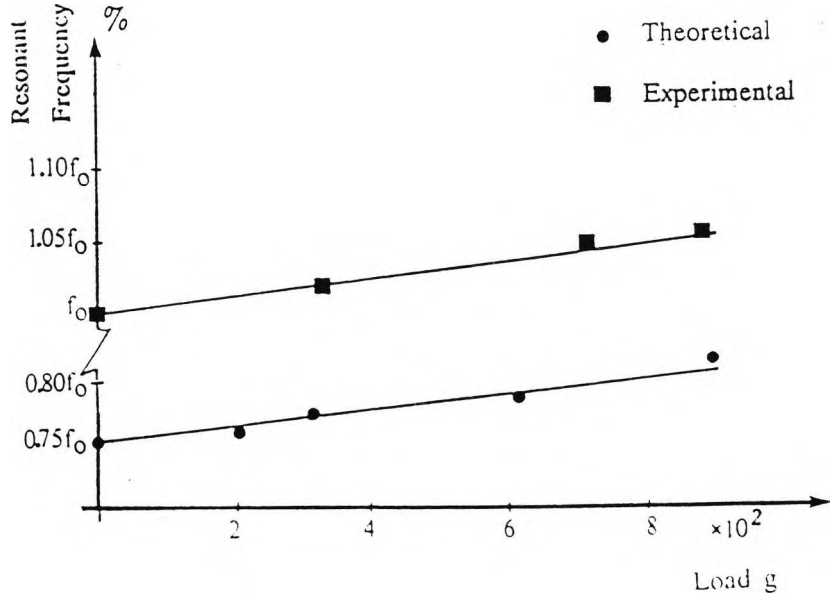


Figure 2.8. Predicted frequency/load relationship of a prototype fibre optic based pressure sensor.

2.5 Conclusions

In this chapter a review of the development and use of mathematical models for sensor development has been provided together with an introduction to the finite element method which is now widely used in many branches of engineering. The work on optical sensor modelling highlighted the **development** and **validation** of mathematical models in a relatively new area whereas the diaphragm and capsules work highlighted the use of developed and validated models for **design** in a well established conventional area. The particular success of the latter area is that capsules and diaphragms can now be designed in about one tenth of the time and at about one tenth of the cost compared with the largely empirical methods used previously. Such CAD pressure sensors also have improved accuracy, typically 0.1% total error compared with 0.2% to 0.5% using conventional methods.

The description, development and use of mathematical models applied in this chapter to industrial pressure metrology leads naturally to tackling the problem of pressure standards metrology in the following chapters.

CHAPTER 3

Mathematical Modelling of Pressure Balances

3.1 Introduction

Having reviewed the area of mathematical modelling of pressure sensors and introduced the finite element technique in the last chapter, this chapter introduces the pressure balance, reviews previous modelling work and then proceeds to explain the theoretical approach adopted towards the modelling of pressure balances. The approach is implemented in the form of two programs: the **F**inite **E**lement **A**nalysis (**FEA**) program which is used to characterise a pressure balance and the **P**ressure-**V**iscosity-**E**lasticity (**PVE**) program which uses the FEA program-generated data to allow simulation of any pressure balance (oil or gas operated) inclusive of both elastic and viscosity effects.

3.2 The Pressure Balance

A pressure balance (also known as a piston gauge or dead-weight tester) comprises a piston-cylinder assembly where the piston is engaged into a virtually sealed-base cylinder while the piston has a free rotational motion to ensure co-axiality and to reduce friction [1]. The schematic diagram for a simple type of a pressure balance is shown in figure 3.1. The piston and cylinder, both of a known cross sectional area, are separated by an initial clearance gap along a common distance of a fixed length known

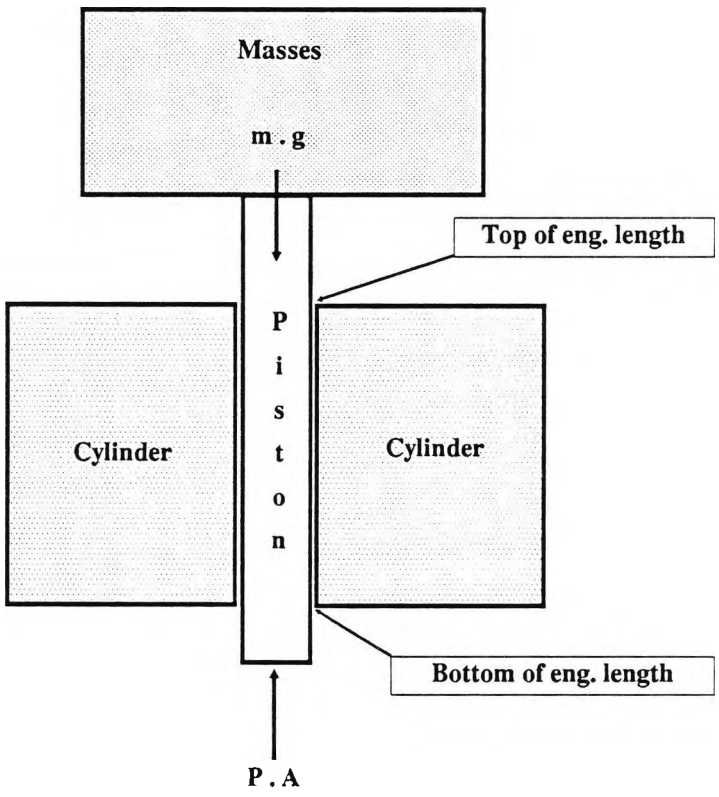


Figure 3.1. Schematic of a simple geometry piston-cylinder assembly (pressure balance).

as the engagement length. The piston often protrudes beyond the engagement length at the top and the bottom of the cylinder. Generally, pressure balances can either be oil or gas operated . In the oil type of balances, a transmitting fluid (lubricating oil) of known viscosity fills the base of the cylinder and the surrounding surface of the piston, retaining the piston in a floating position. On the other hand, in the gas operated pressure balances (assemblies using compressible fluids), air or gas (e.g. helium or nitrogen) replace the oil as the lubricating fluid and they often incorporate the facility to evacuate an ambient space around the piston-cylinder assembly.

Most of the available oil operated pressure balances operate in a mode known as the "gauge" mode whereas the gas operated balances usually operate in either "gauge" or "absolute" mode. A pressure balance generally measures the differential pressure across the top and bottom ends of the engagement length. In the gauge mode, the pressure at the top end of the engagement length is the ambient atmospheric pressure while in the absolute mode it is nominally zero.

The principle of operation of a pressure balance can be expressed by the basic equation for the pressure generated;

$$P = \frac{m \cdot g}{A} \tag{3.1}$$

which indicates a balance of the gravity force, mg, against the pressure generated force, PA, on the piston. The determination of the effective area, A, is of fundamental importance to the operation of the balance.

One very significant fact is that A is not constant because the piston and cylinder both distort under pressure. With A_o being the undistorted (nominal) area, the distortion coefficient λ is defined by the equation

$$A = A_o (1 + \lambda P) \qquad (3.2)$$

and the constant λ can be determined by experimental or theoretical means. Typical values for λ can be between -2 and 4 ppm (parts per million) per MPa which becomes a significant and dominating correction term particularly for high pressure balances.

With regard to other designs of pressure balances, besides the simple design introduced, there are three other designs for high pressure balances. These are summarised as (Lewis et al [14]):

a. A pressure balance similar to the simple design but one in which the load is applied to the piston via an additional piston known as the auxiliary piston. This arrangement (shown in figure 3.2a) reduces the bending moment effects on the piston.

b. A re-entrant pressure balance (shown in figure 3.2b) where the generated pressure is allowed to act on the external surface of the cylinder as well as the base of the piston and along the engagement length.

c. A controlled-clearance pressure balance allowing the application of variable pressure to the external surface of the cylinder, aiming at the provision of some control on the gap along the engagement length. The schematic of this type of pressure balance diagram is shown in figure 3.2c.

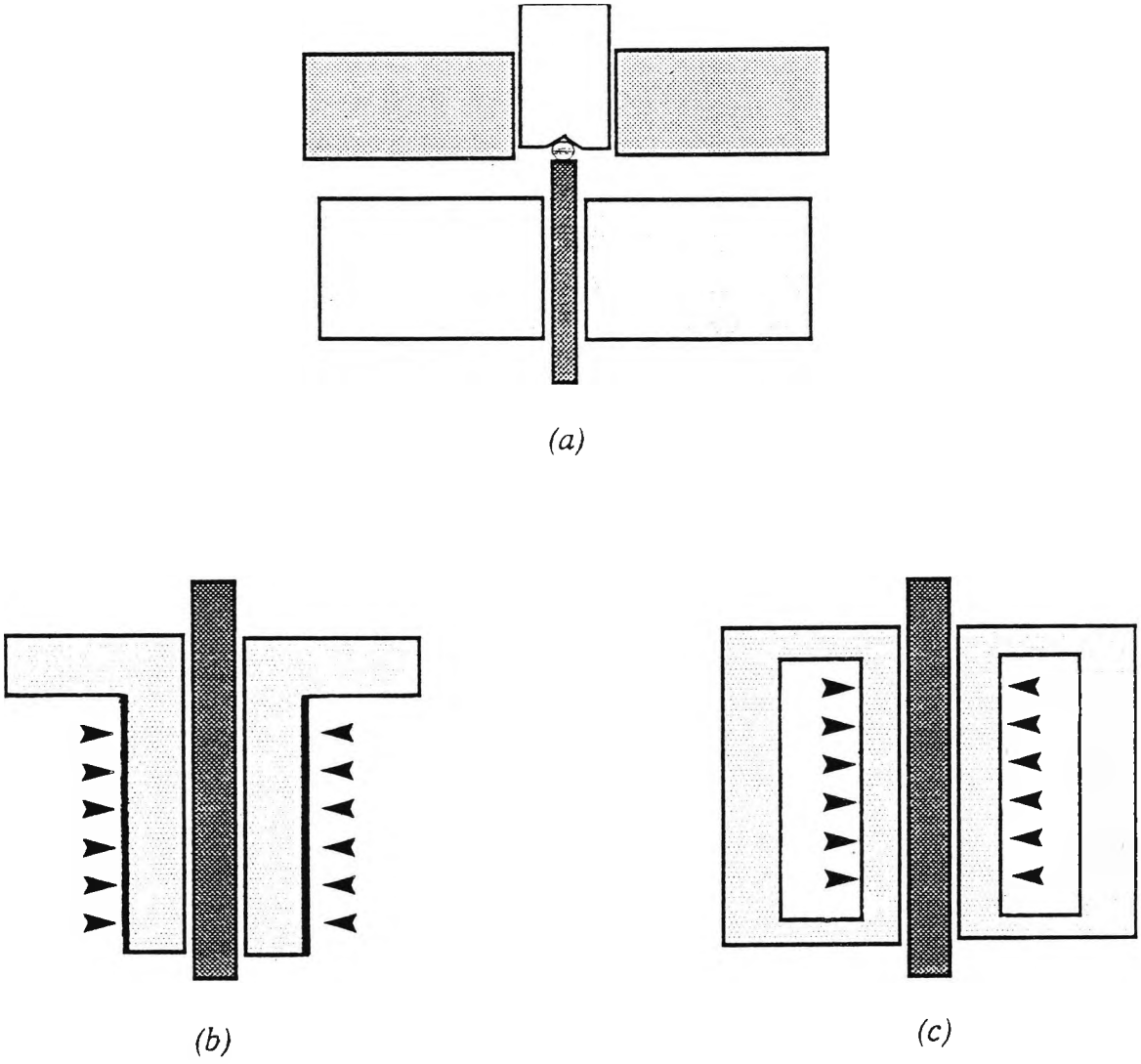


Figure 3.2. Schematic of other pressure balance geometries. (a) With auxiliary piston, (b) A re-entrant design and (c) A controlled-clearance design.

In designs **b** and **c,** the gap between the piston and cylinder is contracted over part of the engagement length, thus reducing the fluid leakage.

In general, the pressure balances are used to form the basis of the high pressure metrological scale within National Standards laboratories [1]. This scale extends up to and beyond 1000 MPa and is covered by a series of balances. For each balance, it is of great significance to have an acceptable level of confidence regarding uncertainties. The procedure in establishing a scale is to construct the lowest primary pressure balance at the highest possible accuracy using diametral measurements of the piston and cylindrical bore (see, for example, Peggs et al [15]). This balance will in turn be used to determine the accuracy of the next higher operating pressure range balance and so on to form the basis of the high pressure scale. This is implemented by a comparison procedure which involves connecting a pressure balance assembly directly to another one with a common reference level, both subject to a common applied pressure. This procedure is simply known as "balancing" or "cross-floating" and it allows the calculation of the effective area of the higher balance in terms of that of the lower one.

3.3 Determination of the Effective Area

The effective area for an ideal balance assembly (i.e. one in which the piston and cylinder are each straight, coaxial and free from taper) can be expressed as the arithmetic mean of the cross sectional areas of the piston and cylinder. For non-ideal cases, Dadson et al [1] have derived a formula for the effective area. These cases arise from either geometrical imperfections of the balance components and/or elastic distortions.

A general formula for the effective area, A, derived from equation 23 of reference [1] (with some notational differences*) and used subsequently in this work is @:

$$A = \pi r_o^2 \left\{ 1 + \frac{h_o}{r_o} + \frac{1}{r_o P} \int_0^L \left[u + U \right] \frac{dp}{dy} dy \right\} \qquad (3.3)$$

$r_o = r_p + y_p(L)$ and $h_o = g + y_c(L) - y_p(L)$,

$u = y_p(y) - y_p(L)$ and $U = y_c(y) - y_c(L)$

where, as shown in figure 3.3,

r_p is the radius of the *undistorted* piston,

g is the *initial* clearance gap between piston and cylinder,

r_o is the radius of the *distorted* piston at the *lowest extent* of the engagement length (bottom end of the engagement length),

h_o is the *radial gap* at the *same level*,

y_p is the piston radial distortion at *any point* along the engagement length,

y_c is the cylinder radial distortion at *any point* along the engagement length,

u, U are the *radial distortions* with respect to the bottom of the engagement length, for the piston and cylinder respectively,

P is the balance operating pressure (i.e. the pressure difference between the bottom and the top of the engagement length),

L is the engagement length,

p is the pressure at *any point* along the engagement length *referenced* to the pressure at the top of the engagement length (i.e. $p(0) = 0$ and $p(L) = P$).

t is the distorted clearance gap at *any point* along the engagement length ($t = g + y_c - y_p$).

Note: The nominal (undistorted) area, A_o, is obtained from equation 3.3 putting all distortions equal to zero, therefore $A_o = \pi \cdot r_p^2 \{ 1 + (g/r_p) \}$. Also, equation 3.3 has been specialised to the case of elastic distortions.

To determine the quantities y_p and y_c in terms of the applied pressure, evidently, the situation is complicated by the fact that the pressure distribution itself is dependent on the radial distortions in a manner which is in turn dependent on the variation with the pressure coefficient of viscosity of the transmitting fluid. A detailed analysis by Dadson et al [1] shows that a simplification can be made if it is assumed that radial

* In eq. 23 of ref. [1], p is the absolute pressure. Also, the integrations are in opposite directions leading to a sign difference for the integration term.

@ The origin of the equation involves the balance of forces on the piston (downward force, mg, and three pressure generated forces: on the base, on the flanks and a drag force on the flanks due to fluid motion).

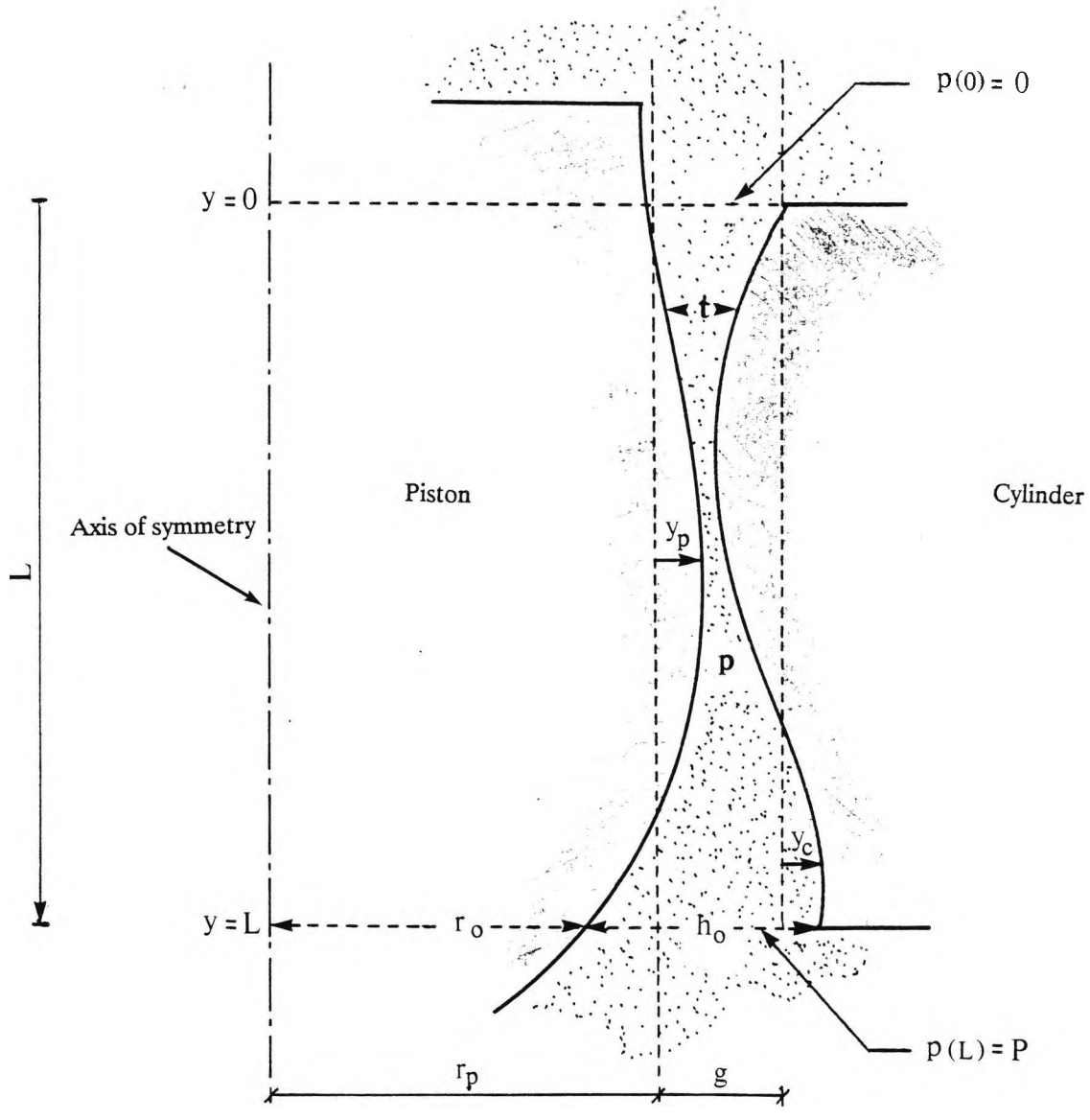


Figure 3.3. Piston and cylinder geometrical variables.
(Notes: 1. t, y_p, y_c *and* p *vary along the engagement length. 2. The undistorted piston and cylinder are assumed to be of perfect geometry, therefore* r_p *and* g *are constants.)*

distortions (y_p and y_c) are each proportional to the actual applied pressure at the same position. As a result, it can be concluded that the distortion coefficient, λ, is:

a. independent of the applied pressure (effective area is only a linear function of the applied pressure).

b. independent of the rheological properties of the transmitting fluid.

c. a combination of two separate parts depending on the elastic properties of both the piston and the cylinder.

As shown by Peggs et al [15], for a low pressure primary pressure balances (e.g. up to 5MPa), the determination of the distortion coefficient with great accuracy is neither practicable nor necessary. However, distortion effects are of great significance as the operating pressure of the pressure balance is increased. Then in practice, for higher pressures, the distortion coefficient is determined accurately by cross-floating with another balance using the similarity method (see reference [1] for details of these methods).

With regard to the sources of uncertainties in the use of pressure balances, in general, these sources may be classified into two groups- intrinsic and extrinsic. The intrinsic ones arise from the calculation of the load forces and the pressure actuating the balance at a chosen level. However, it is common that the quantity of direct interest may well be the pressure at an external site in a system connected to the balance (e.g. a secondary pressure gauge to be calibrated or where a pressure phenomena is to be observed). The latter is known as the extrinsic source(s) of uncertainties [1]. The major intrinsic source of uncertainties would be in the determination of the effective area. This includes:

i. Dependence upon temperature, which is merely determined by the thermal expansion coefficient of the balance components (piston and cylinder).

ii. Dependence upon the applied pressure which arises from the effect of the elastic distortion of the piston-cylinder assembly and the departure from the perfect cylindrical form of the piston or the cylinder. Added to that is the effect of the dynamic viscosity of the transmitting fluid which, at high pressures, affects the pressure distribution along the engagement length and in turn will alter the elastic distortion of the balance components.

iii. Dependence upon the method of attachment which incorporates, however carefully calibrated, the pressure seal and other complementary components of the pressure balance (screw threads, other fixing devices etc.) being close to (or even overlapping) the region of engagement of the piston and cylinder, the insufficiently thick cylinder walls and finally even the degree of the tightness of the closure of the system.

A summary of various sources of uncertainties is well documented by Peggs et al [15] for a low primary pressure balance (up to 4 MPa). It can be seen from Table 6 of reference [15] that the major contributors to the sources of uncertainties are the temperature effects (20 ppm/ ^{o}C) and the distortion coefficient (4.1 ppm/MPa) for the low pressure balance studied. Since temperature can be controlled to a fraction of a degree (0.1^{o}C), the temperature effect becomes insignificant for higher pressure ranges leaving only the distortion coefficient as the major source of uncertainty.

3.4 Review of Previous Work

The pressure balance has been researched and documented by several workers over many years. Dadson et al [1] provide a comprehensive review which includes over a

hundred references indicating the importance of the pressure balance in measurement. The main thrust of the research and the amount of work done were aimed for the accurate determination of the distortion coefficients. Dadson has shown, on quite general theoretical grounds, that for balances of simple geometries, the distortion coefficient is a constant (i.e. equation 3.2 is valid without any higher order terms). A recent meeting (1988) of the "High Pressure" group was held in Paris with the purpose of presenting an update survey by metrological National Laboratories concerned with high pressure measurement [16]. Molinar et al [2] quoted that a recent comparison between the work of thirteen National Metrological Laboratories, showed that the determining factor limiting the accuracy of pressure balances is an imperfect knowledge of the coefficient of elastic distortions in the piston-cylinder unit (the balance assembly).

A theoretical analysis reported by Wisniewski et al [17] used the Finite Element (FE) method to model a pressure balance (operating up to 300 MPa) assuming a linear and parabolic distribution along the engagement length. The distortion coefficient determined using the elastic distortions from the two pressure profiles seemed to agree with that of Dadson et al [18]. However, the analysis in reference [17] is incomplete due to the fact that pressure distribution along the engagement length can be anything between a severe drop at the top end to a severe drop at the bottom end of the engagement length. This is due to the order of magnitude of the elastic distortion compared to the initial gap as well as the effect of the viscosity variation due to the pressure viscosity dependence. The latter effect has not been taken into account in the analysis by Wisniewski et al [17]. Similar work has been reported by Klingenberg [19] but using analytical methods to determine the elastic distortions of pressure balances. This assumes a quadratic term in the pressure distribution along the engagement length of a high pressure balance. Since the effect of the dynamic viscosity is not taken

into account, as the pressure balance is one operating at high operating pressure, the quadratic term may not be sufficient to represent the actual pressure distribution which, in turn, will generate the elastic distortions used in the calculation of the effective area and the distortion coefficients.

However, the effect of pressure-viscosity variations is analysed by Stuart [20] for different pressure ranges and two commonly used oils. In the latter analysis, an iterative method for the calculation of the actual pressure distribution (in the case of high pressure balances) is used, but the analysis is based on the assumption that the clearance between the piston and cylinder along the engagement length will remain constant. This implies that the effect of the elastic distortions is not already considered. In fact, Stuart suggests as a conclusion, that an alternative method would be to incorporate both the elastic distortions and the viscosity effect in an iterative analysis. This means that for an initial assumption of a pressure distribution (e.g. a linear one) elastic distortions will generate a specific clearance profile. For this profile, the effect of the viscosity is taken into account, resulting in a new pressure distribution. At this stage, the elastic theory has to be applied again (but with a new pressure profile) to determine a new clearance (gap) profile and so on, iteratively, until a small (e.g. less than 1%) change is observed either in the clearance or the pressure profiles along the engagement length. This will lead to the more accurate determination of the distortion coefficient.

The above procedure has been implemented by Molinar et al [2] using analytical methods. This requires the need for the repetition of the procedure concerned with the determination of the elastic distortions every time a new pressure distribution along the engagement length is to be found. This may require several hours for a particular balance and more importantly, the requirement for the "same" several hours,

for any repeated analysis for the "same" balance (such as a different initial gap, different oil etc). Further, the method adopted by Molinar [2] is based on an analytical approach for the determination of elastic distortions which is likely to be a limitation in case of modelling pressure balances of complex geometries. The latter limitation can be reduced significantly if the Finite Element (FE) method is employed (also suggested by Stuart [20]) due to the flexibility provided by the nature of the method itself.

It is interesting to note that the use of the (FE) method may overcome (probably to a large extent) the difficulty of modelling pressure balances of a complex geometry but it (i.e. the FE method) would still require a considerable time which may not be, at least in some cases, very different from that required for the analytical approach of Molinar [2]. That is for the same reason, this being the need to carry out the FE simulation every time a new pressure distribution is found and also, for every further exploration of the same balance.

A better approach, implemented in this work, would be to **characterise** the elastic distortions of any given balance in a form of some "base" data. The provision of this "base" (also described later as the "unit" load data) will inhibit the requirement for the repetitions of the FE analysis in an iterative approach.

With regard to the uncertainties in the measurement of the elastic distortions, measurement reported by Legras [21] claims that the uncertainties in λ can be estimated to within 0.05 ppm/MPa for a pressure balance operating up to 200 MPa. Legras indicates that with such precise knowledge of λ, uncertainties in the highest pressure generated (1000MPa) can be as low as 80 ppm.

Measurement of pressure profiles along the engagement length of pressure balances have been performed by Welch and Bean [22]. These show a far from linear distribution along the engagement length (as is commonly assumed). From the pressure distribution, also confirmed by theory, Welch and Bean calculated the distortion coefficient for 21 MPa oil balances. To do this, they calculated an "effective" pressure along the engagement length by considering the point at which the pressure gradient, which they call the "downbreak" point, changed rapidly. Subsequently, Welch and Bean determine the distortion coefficient using the "downbreak" pressure point which appears as P_a/P_s in equation 2 of their paper. This equation is of a questionable validity. When $P_a/P_s = 0.5$, the equation agrees with a form obtainable from equation 104 of Dadson et al [1]. In conclusion, it appears that there is no sound theoretical reason for using the formula except at $P_a/P_s = 0.5$. Indeed if the formula and method adopted by Welch and Bean were used in general, then the distortion coefficient would depend on the pressure profile (i.e distribution) which can change markedly during the operation of a high pressure balance as is shown later in the present work.

3.5 Theoretical Approach

In computing the effective area of a pressure balance, the pressure distribution along the engagement length is required as well as the elastic distortions of the piston and cylinder (equation 3.3). For high pressure oil operated balances, the pressure-viscosity dependence becomes a significant factor in determining the pressure distribution which in turn determines the elastic distortions. Therefore, the requirement is to incorporate both the elastic and viscosity effects in an appropriate model. The novelty of the model to be adopted relies upon the characterisation of any given pressure balance for its elastic properties by computer generated "base" data for further analysis incorporating viscosity effects. The characterisation avoids the requirement for high processing time as well as providing a general tool for treating complex geometries.

Dadson et al [1] has documented the theory and practice of the pressure balance and this has been well researched. Elasticity theory is applied to a common piston-cylinder and it is concluded that the effective areas are only a linear function of the applied pressure, characterised by the distortion coefficient and the latter is independent of the rheological properties of the pressure transmitting fluid. Certain assumptions have gone into the theoretical analysis, the main ones being:

a. that Saint-Venant's principle applies so that cylinder and piston end-effects can be neglected and

b. the pressure profile along the engagement length of the assembly is assumed to be reasonably smooth.

However, when the pressure distribution changes rapidly at either end of the engagement length both assumptions are simultaneously invalid. This situation can arise either when the gap under distortion becomes relatively small at the top end of the engagement length (leading to a rapid pressure drop at this end) or when the pressure-viscosity dependence of the fluid is very severe (leading to a rapid pressure drop at the bottom end). Indeed a paper by Stuart [20] shows that by iterative computer simulation, for a *constant gap*, 90% of the pressure can drop along the bottom 10% of the engagement length, for a pressure balance operating at 800 MPa with a high temperature lubricating oil as the pressure transmitting fluid. The initial clearance gap (initial gap) in high pressure balances is often kept as small as possible (typically 0.5 μm or less) and then under operating pressures the distorted gap could be far from constant with the possibility of a large pressure drop occurring at the top end of the engagement length. This is so because elastic distortion effects tend to open up the bottom of the engagement length and, possibly, close up the top end.

There is therefore a need for a better theoretical understanding of pressure balances in order to compute distortion coefficients for any pressure balance to determine if there are any deviations from the idealised situation described in equation 3.2. The approach adopted is to use the finite element method. This allows the evaluation of the distortion of the piston and cylinder as two separate problems for a given pressure distribution along the engagement length. The problem is that the actual pressure distribution will depend on the gap profile along the engagement length and also (particularly at high pressures) on the pressure viscosity dependence of the fluid. An iterative approach is therefore needed.

One approach would be to assume a particular pressure profile (e.g. linear) along the entire engagement length and find, using an FE program, the piston and cylinder distortions to obtain the gap profile along the engagement length. With this profile the one-dimensional laminar flow problem is then solved in an iterative manner (to incorporate viscosity-pressure dependence as done in reference [20] except now for a variable gap) to obtain a new pressure profile. The procedure is then repeated by re-running the FE program iteratively until there is convergence for the pressure profile. In principle the whole procedure could be automated using, say, the Newton-Raphson method applied to a particular pressure balance with a known initial gap and pressure transmitting oil. A severe problem is that the whole procedure would have to be repeated if the same balance was to be analysed with a different initial gap and/or a different oil. Given that it may take up to half an hour of computing time for a single FE "run" to obtain the distortion of the piston and cylinder for a given pressure, and perhaps 10 or more iterations for final convergence, it was therefore essential to find a more general and efficient approach.

The approach adopted was to characterise a given piston-cylinder assembly through the application of known "unit" loads to both the piston and cylinder. A general purpose-built **F**inite **E**lement **A**nalysis (**FEA**) program was developed and this program is run for each "unit" load and the distorted profile data along the entire engagement length for the piston and cylinder is kept as a data file. The unit loads are moved along the engagement length of the assembly and a series of data files are constructed for the piston and cylinder. Typically these may be 11 for the piston and 11 for the cylinder, thus totalling 22 runs of an FE program. The method relies on the superposition principle of linear systems which is valid for pressure balances since Hooke's law applies. With this principle the distortions for an arbitrary pressure distribution could be reconstructed and it becomes a much simpler exercise iteratively to obtain the final converged pressure and gap profiles along the engagement length. From the latter, the effective area and hence distortion coefficient can be calculated. New conditions (e.g. a change in initial gap or change in the exponent term for the oil viscosity-pressure relationship) could be simulated interactively within minutes. The theory behind the method is described in more detail in section 3.6. The program developed is known as the PVE (**P**ressure- **V**iscosity-**E**lasticity) program as it is an extension of what may be called the PV (**P**ressure-**V**iscosity) program developed by Stuart [20], since the latter does not consider elastic distortions. The following section describes the detailed theoretical approach employed in the **PVE** analysis model and which operates on the "unit" load data generated by the **FEA** (described in section 3.10). The models developed were tested on the UK National Physical Laboratory (NPL) series of balances details of which are supplied in Appendix (A).

3.6 Development of the Pressure-Viscosity-Elasticity (PVE) Program

This section describes the PVE iterative program which accommodates both low and high pressure balances, whether oil or gas operated. It operates on the calculated

elastic distortions characterising a pressure balance obtained from the FEA program. At an early stage of the work, piston and cylinder distortions were obtained (using the FEA program) for the NPL series 100 pressure balance at 1 MPa operating pressure. These distortions were used by a program documented by Arbani [23] for the determination of the effective area. The distortion coefficient for the series 100 NPL pressure balance was found to be 3.12 ppm/MPa. To calculate distortion coefficients at other pressures, the method used by Arbani was simply to multiply the piston and cylinder engagement length distortions by the required pressure value (in MPa) and then re-run the effective area calculation program with the new distortions. The linearity assumption of elasticity theory is, of course, invoked in doing this. The results are certainly expected to be wrong for high pressure balances when viscosity changes become significant but they may also be incorrect for low pressure balances. The reason for this being that the 1 MPa "starting" distortions were obtained by applying a linear pressure distribution along the engagement lengths. This linear pressure distribution is only valid if the gap along the engagement length is constant. Although the initial gap along the engagement length is taken as constant, the distortion itself will change this, if it is comparable to the initial gap. That this effect is significant, is seen in the variation of the distortion coefficient for the series 100 results taken from appendix K of reference [23].

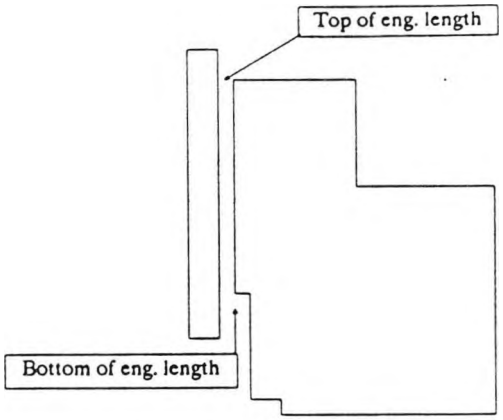
Pressure (MPa)	**Distortion Coefficient** (ppm/MPa)
0.5	3.23
1.0	3.12
1.5	3.16
2.0	3.09
2.5	3.10
3.0	3.05
3.5	3.04
4.0	3.00
4.5	2.95
5.0	2.95

Table 3.1. Distortion coefficients for the NPL Series 100 Pressure Balance

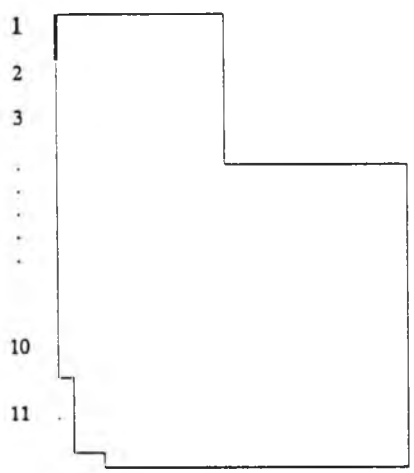
The results show a decrease in λ of several per cent over the range, which violates the assumption that λ is constant in the equation 3.2.

The decrease in λ was even greater for the series 200 balance (25%). To remedy this, an iterative approach could be adopted to determine the correct pressure profile and a **PE** (**P**ressure-Elasticity) program could be developed for low pressure balances where viscosity effects are insignificant. A better approach is to develop the more general PVE (**P**ressure-**V**iscosity-Elasticity) program which will accommodate both high and low pressure balances. Such a program will "automatically" model low pressure balances and indeed indicate the departure point between low and high pressure balances by perhaps looking at the departures, if any, from equation 3.2.

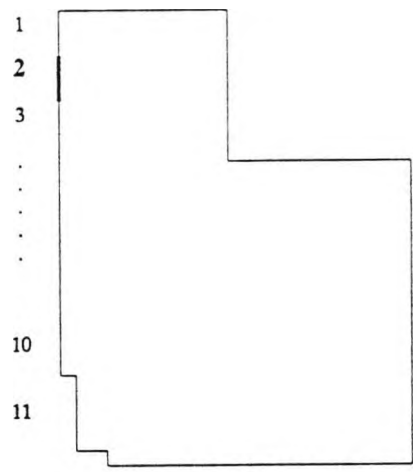
The first requirement in developing a PVE program is to characterise a piston-cylinder assembly by unit load data. A typical piston- cylinder arrangement is shown in figure 3.4a. The engagement length (of say, the cylinder) is divided into a number of sections (e.g. 10 as shown in figure 3.4b) and a unit pressure load (1 MPa) is applied to each section in turn. For each load case we find (using the FE program) the radial displacements (distortions) at the centre of each section and form arrays y_{c1}, y_{c2}, ..., y_{c10} each with 10 values. For example, y_{c2} will contain the distortion data (beginning from the top of the engagement) when load case 2 (figure 3.4c) is applied. As the concern is only with changes of pressure that may occur along the engagement length one more array y_{c11} (or y_{cn+1} with n sections along the engagement length) is needed for loadings away from the engagement length. The loading for this, is shown in figure 3.4d.



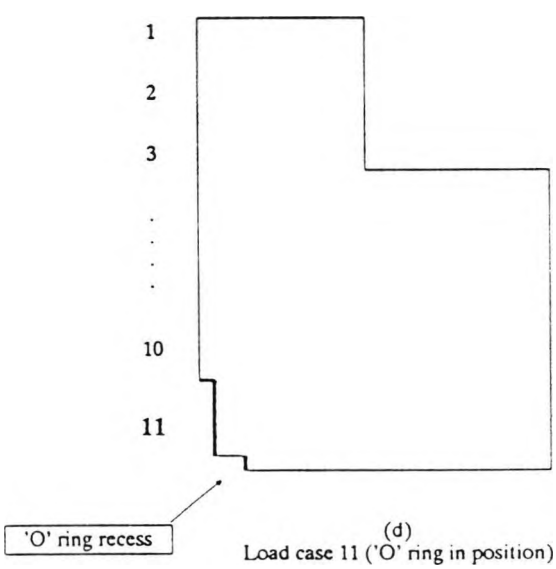
(a)
Typical pressure balance (axisymmetric view)



(b)
Load case 1



(c)
Load case 2



(d)
Load case 11 ('O' ring in position)

Figure 3.4 Engagement length and its unit load disribution.

Then using the superposition principle (assuming n sections), the radial distortion can be found:

$$y_c = \sum_{i=1}^{n+1} a_{ci} \cdot y_{ci}$$

along the engagement length for any pressure distribution where the coefficients a_{ci} are obtained by averaging the pressure distribution as shown in figure 3.5.

For example,

$$a_{c4} = \frac{p_a + p_b}{2}$$

Of course the continuous distribution is being approximated in a piecewise stepped way but this is a reasonable approximation with an adequate number of sections, n.

An identical procedure can be adopted for the piston remembering the engagement length for the piston starts and ends away from the two ends of the piston. This leads to a distortion profile,

$$y_p = \sum_{i=1}^{n+1} a_{pi} \cdot y_{pi}$$

Thus with 10 sections there is a need to generate, once and for all, 22 arrays (i.e. run the FE program 22 times) for a particular cylinder support condition. If the cylinder support condition is changed (e.g. from sliding to clamped) only the cylinder runs will need to be repeated.

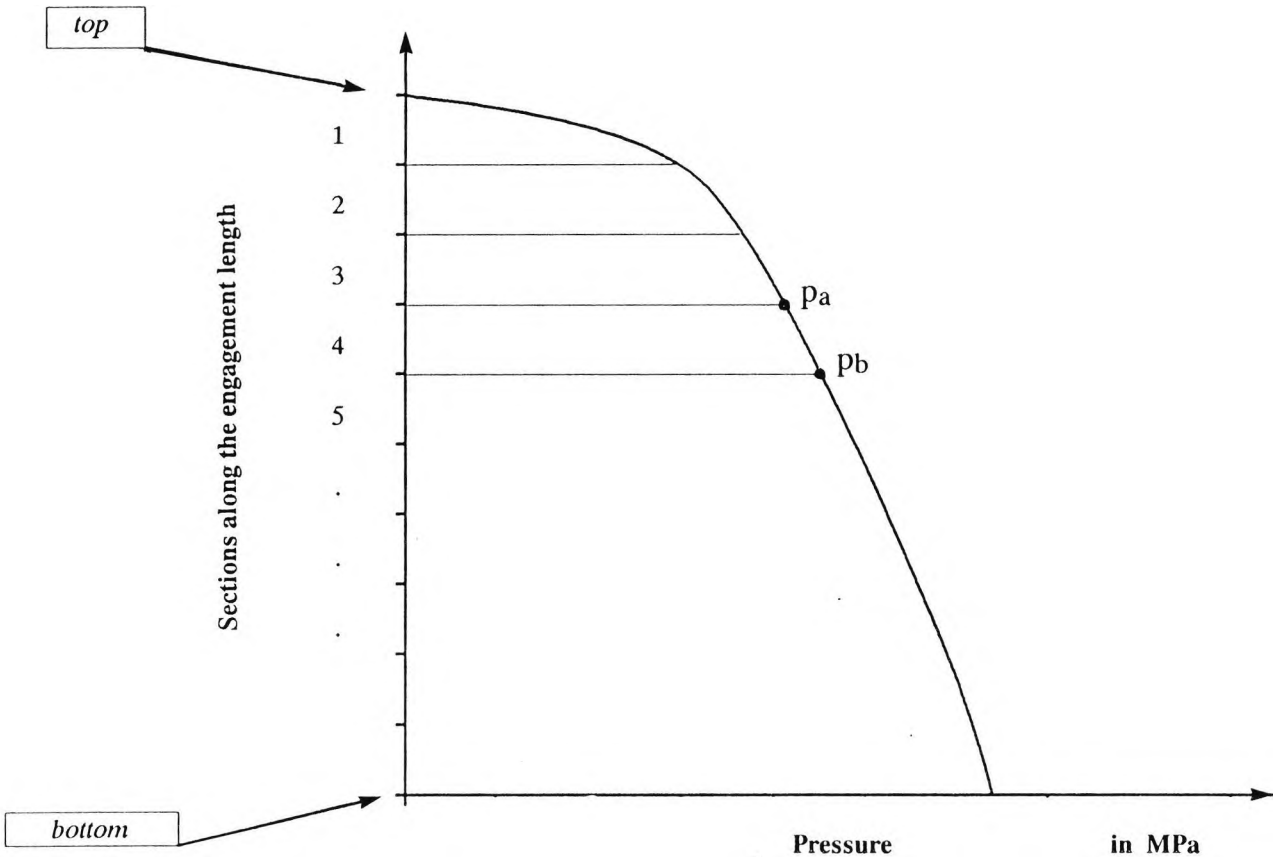


Figure 3.5. A typical pressure profile and its use for calculating a_c *coefficients.*

Clearly if a constant (1 MPa) pressure profile along the engagement length is applied then all the values of the **a**'s ($\mathbf{a}_p$'s and $\mathbf{a}_c$'s) will equal 1. For a linear pressure distribution the array of **a**'s ($\mathbf{a}_p$ or $\mathbf{a}_c$) will be (0.05, 0.15, 0.25, 0.35, ..., 0.95, 1.0). Note the last value of the array is always 1 (or the applied pressure in MPa in the general case).

The procedure adopted in the PVE program is as follows:-

1. Obtain **a**'s by assuming a known pressure profile along the piston-cylinder engagement length (a linear one is assumed).

2. Find the radial distortions yp, yc and hence (knowing the initial gap) the gap profile along the engagement length.

3. With this gap profile assumed "frozen", use the **PV** (**P**ressure-**V**iscosity) iterative procedure to find a new pressure profile.

4. Calculate new **a**'s as indicated earlier, then repeat (iteratively) step 2 to 4 until convergence for the pressure profile is obtained.

5. Post process to display converged pressure and gap profiles and calculate the distortion coefficient.

The basis of the **PV** (**P**ressure-**V**iscosity) procedure is the application of a one-dimensional laminar flow theory equation for the flowrate F through the annular gap between piston and cylinder given by [20]*:

$$F = \frac{p}{L}\ \frac{1}{\eta}\ \frac{\pi R t^3}{6} \qquad (3.4)$$

where p is the pressure difference between the ends of the gap,

* The π in equation 3.4 is missing in reference [20], but this makes no difference to subsequent results since it is absorbed in the constant (k) introduced in equation 3.6.

L the length of the gap (engagement length),

η the coefficient of dynamic viscosity, assumed pressure independent,

R the mean of the radii of the piston and cylinder { $R = r_p + 0.5\, h_o$, (see figure 3.3)},

and t the width of the gap between the piston and cylinder.

In practice the viscosity can change by orders of magnitude along the engagement length for high pressure balances. However, equation 3.4 can still be applied but to a very small section or layer along the engagement length because the gap t is very small compared to the engagement length (typically a ratio of 1 to 25000). So, even for 500 layers along the engagement length the ratio of gap to the layer length is 1 to 50 and therefore, along such a layer length, t and η can be considered constant and therefore equation (3.4) becomes:

$$F = \frac{\Delta p}{\Delta L} \frac{1}{\eta} \frac{\pi R t^3}{6} \tag{3.5}$$

where now η is the mean (pressure dependent) viscosity along a layer of length ΔL and Δp is the pressure drop across the layer. By flow continuity, F must be a constant. Also the mean radius can be assumed constant, and so equation (3.5) can be rearranged as:

$$\Delta p = \frac{F \Delta L}{k} \frac{\eta(p)}{t^3} \tag{3.6}$$

where $k = \pi R/6$ is a constant. If the entire engagement length is divided into N layers of equal length ΔL, the pressure at the nth layer from the top of the engagement length is just the sum of all the individual pressure drops. Denoting this as p_n, equation 3.6 may be written as:

$$p_n = \sum_{j=1}^{n} \Delta\, p_j = \frac{F\Delta L}{k} \sum_{j=1}^{n} \frac{\eta_j}{t_j^3} \tag{3.7}$$

where η j and tj are values for the jth layer. The value ηj depends on the pressure at the jth layer pj, and is found from a given pressure viscosity relationship:

$$\eta = \eta\,(p) \tag{3.8}$$

and hence for the jth section,

$$\eta\, j = \eta\,(pj) \tag{3.9}$$

In an iterative procedure pj will be taken from a previous iteration.

Defining

$$S_n(\mathbf{p}') = \sum_{j=1}^{n} \frac{\eta_j\,(p_j')}{t_j^3} \tag{3.10}$$

where **p** denotes the array of pressures from a previous iteration, it is found that:

$$p_n = \frac{F\,\Delta L}{k}\; S_n(\mathbf{p}') \tag{3.11}$$

At the bottom of the engagement length, where n = N, the pressure p is fixed at the balance operating pressure so pN is a constant (equal to the balance operating pressure, P). The constant term FΔL/k can then be found and (3.11) becomes:

$$p_n = \frac{p_N}{S_N(\mathbf{p}')} S_n(\mathbf{p}') \qquad (3.12)$$

Calculating p_n for all layers results in the current pressure array **p**. For convergence, **p** and **p′** must be the same. The procedure for the **PV** program is then summarised as:-

a. Assuming a linear pressure profile along the engagement length, an initial array **p′** is constructed for the pressure along the N equal sections.

b. Using equation (3.10), $S_n(\mathbf{p}')$, (n = 1 .... N) is computed, and hence from (3.12) the current array **p** can be found.

c. Test for convergence (Is **p** = **p′**?). If not, reset **p′** to **p** and repeat steps b and c (i.e. iterate) until convergence occurs.

The NPL series 400 pressure balance was chosen to test the above procedure. This required the generation of the "unit" load data which in turn requires the application of the FEA program (section 3.7 and 3.9.2).

A typical convergence for the pressure profile was assumed when the elements of the two arrays **p** and **p′** do not differ, on average, by more than 0.01%. A typical number of layers chosen is 500.

The oil used in all the NPL pressure balances is a high temperature lubricant, di-2-ethyl hexyl sebacate, a synthetic ester of molecular weight 426. This has advantage over commonly used mineral oils in remaining a liquid in the range up to 800 MPa. Its viscosity dependence is taken from [20]:

$$\log_{10}\eta + 1.2 = (\log_{10}\eta_o + 1.2) \cdot (1 + p/200)^z \qquad (3.13)$$

with η_o = 21.1 mPa.s and the viscosity exponent term z = 0.55, pressure p in MPa. The viscosity ratios at the maximum operating pressures for the series 100 to 400 balances* are then easily computed as 1.08, 1.37, 3.26 and 55.6 respectively, indicating a marked change only for the series 400 balance. For a balance operating over the full possible range for the oil (up to 800 MPa) the change is an astronomically large value of 3900.

With 500 layers and a 0.01% error on pressure convergence, the PV subprogram of the PVE program converged in the order of 1 minute for the series 400 balance operating at maximum pressure (320 MPa). Convergence of the pressure profile at the 0.1% to 1% error level in the PVE program took typically ten iterations and therefore a total run time of approximately 10 minutes. These times were for the program running on a Whitechapel MG-1 workstation. There is an improvement by approximately a factor of ten in time when the PVE program is run on the Sun/SPARC workstation.

High pressure balances now operate up to, and in some cases beyond 1000 MPa. Therefore it was essential to test the PVE program for such higher pressures. This was done with the NPL series 400 balance and it was found that convergence could not be obtained at pressures greater than 500 MPa. An improved method which does converge, was then adopted as explained in the following section.

* The maximum operating pressures are assumed to be: series 100; 5 MPa, series 200; 20MPa, series 300; 80MPa and series 400; 320MPa . Further, the viscosity ratios are with respect to ambient pressure (η_o the value at the top of the engagement length).

3.7 Improvement to PVE Convergence

The flow equation (3.5) can be rewritten in differential equation form rather than the difference form given, so:

$$F = \frac{dp}{dy} \quad \frac{1}{\eta(p)} \quad \frac{\pi R t^3}{6} \tag{3.14}$$

where y is the distance along the engagement length (from its top). The method adopted by Stuart [20] and extended in the present work, as described in the previous section, was to re-arrange the above equation as:

$$dp = F \frac{\eta(p)}{k t^3} dy \tag{3.15}$$

where $k = \pi R/6$

p is then obtained by integration. The integration cannot be done explicitly since η is a function of p. Consequently, the **PV** iterative method was developed, where essentially values for p used in η are obtained from a previous **PV** iteration and the gap, t, was variable with y but "frozen" for the entire **PV** iteration.

If equation 3.14 is rearranged and integrated, then

$$\int_0^{\dot{y}} \frac{6F}{\pi R} dy = \int_0^p \frac{t^3}{\eta(p)} dp \tag{3.16}$$

then the integration (now along the p axis) can be done explicitly since η (p) is known, so no iteration is required.

Letting $6F/\pi R = \alpha$ (a constant) and I(p) equal to the r.h.s. of 3.16,

$$\alpha \cdot y = I(p) \tag{3.17}$$

and α is found from the condition at the bottom end of the engagement length for which $y = L$ and $p = p_N$, so,

$$\alpha = \frac{I(p_N)}{L}$$

and therefore,

$$y = L \frac{I(p)}{I(p_N)} \tag{3.18}$$

I(p) is evaluated numerically as a sum dividing the pressure axis into N equal layers each of step length $\Delta p = p_N/N$. Equation 3.18 will then map p to y with non-equal steps along the y axis in general.

With this method, a modified **PVE** program was produced following steps 1 to 5 (in the previous section) but **no** iteration required for step 3. Also, in step 4, care was required in calculating the new values of **a**'s. The reason being that the **a**'s require **p**'s at mid sections (see for example figure 3.3) on the y axis. The mapping of p to y in equation 3.18 will not produce appropriate mid-section y's so interpolation was required.

3.8 Extension to Gas Operated Pressure Balances

For gas operated balances, compressibility must be accounted for and therefore the mass flowrate must be constant anywhere along the engagement length. Then equation 3.14 can be suitably modified by multiplying both sides by ρ, the gas density. Then $(\rho \cdot F)$ will be the mass flowrate, and I(p) (the r.h.s) has to be modified to:

$$I(p) = \int_0^p \frac{t^3 \rho}{\eta(p)} \, dp \qquad (3.19)$$

For a gas, the density is proportional to absolute pressure, so,

$$\rho = k'(p + p_o)$$

where k' is a constant and p_o is the pressure at the top end of the engagement length (i.e. p_o = atmospheric pressure for gauge and p_o = 0 for absolute mode operated gas balances).

The result is that equation 3.17 is modified to:

$$\alpha' \cdot y = I'(p)$$

where $\alpha' = \frac{6 \rho F}{\pi R k'}$ (a constant) and

$$I'(p) = \int_0^p \frac{t^3(p + p_o)}{\eta(p)} \, dp \qquad (3.20)$$

Consequently, equation 3.18 becomes

$$y = L \frac{I'(p)}{I'(p_N)} \tag{3.21}$$

The **PVE** program was suitably modified and in order to accommodate both gas and oil balances in one program, the term $(p + p_o)$ in equation 3.20 was modified to give $(p + p_o)^{\beta}$ where $\beta = 1$ for **G**as or $\beta = 0$ for **O**il operated balances. Further, for the gas operated balances, $z = 0$ in equation 3.13 since usually there is no pressure viscosity dependence.

3.9 Determination of the Effective Area and the Distortion Coefficients

Having obtained the final converged pressure and gap profile for a balance simulated at its actual operating pressure, the effective area and the distortion coefficient can be calculated simply by using the general formula derived by Dadson et al [1] (equation 23) and mentioned in section 3.3. The distortion coefficient is simply given by:

$$\lambda = \frac{A - A_o}{P} \cdot 10^6 \tag{3.22}$$

in parts per million per MPa (ppm/MPa), where

A is the effective area of the balance,

A_o is the nominal (undistorted) area of the balance and

P is its operating pressure (in MPa).

Finally, the **PVE** program presents graphically the final converged pressure and gap profile alongside a numerical display of the summary of the input data (initial gap, number of layers, oil properties etc). A sample of the input and output displays are given in Appendix B.

For the graphical presentation, the GINO graphics library is used in the **PVE** program to display pressure and gap profiles. In using equation 23 of reference [1], the integration is performed as summation over a (typical) 1000 layers along the engagement length. This approach is very much simpler than using Peggs' effective area program (appendix B of reference [1]). The reason is that Peggs' program has been designed for measured diametrical data. In this modelling work the undistorted balance is assumed perfect in geometry and consequently only about 20 lines of coding were required to calculate the effective area, whereas Peggs' AREA program is several hundred lines of code. A typical **PVE** run will take a few minutes processing time on a Sun/SPARC workstation.

3.10 Development of the Finite Element Analysis (FEA) Program

The FEA program was based on the NAG/SERC library of subroutines for finite element computations. This library was developed originally at RAL with SERC support and is now marketed by NAG Limited as the NAG/FE Library [7]. The FEA program was developed to model any axisymmetric piston-cylinder assembly. Triangular six noded isoparametric elements were used. A separate preprocessor, FEMGEN [24], is used to enter a model and mesh the model with triangular elements. To the geometric and topological data obtained from FEMGEN in the form of a file is added the material property data (Young's modulus, Poisson's ratio), restraint data representing support

(clamping) conditions and pressure loading data. The FEA program developed is then run and an output file of displacements is produced for reading by FEMVIEW [24], a post processor program which allows the display of displaced shape and hence the distortion of the piston (or cylinder). The FEA program has been developed to allow pressure loading on the edges of the triangular elements (in groups or individually). In this way, arbitrary pressure loadings can be prepared before running the FEA program. For example, unit pressure loads on single elements are required along the engagement length for the PVE program whereas it is equally easy to apply a linear graded pressure along the elements of the entire engagement length as a staircase approximation.

In the previous chapter (section 2.3), the NAG/FE library was briefly described. Alterations were made to the LEVEL 1 library to adapt it for the modelling of pressure balances. In this work, example program 1 known as SEGMENT 1.1 (see reference [7]) was initially considered since it can handle the static analysis of plane solids. Modifications made to the code of SEGMENT 1.1 included the way the model geometry and topology are read, coordinate transformation from a plane type of problem (as in SEGMENT 1.1) to an axisymmetry problem (for the pressure balance), the way the loading information (the application of pressure loading on the piston and the cylinder) was applied and finally the modification of output routines to suit the use of the PVE program and the use of FEMVIEW to display the elastic distortions of the balance components. The FEA program, at present, is over 1400 lines of code of which only about 50 lines of it remained unchanged from the original few hundred lines or so of the SEGMENT 1.1 code. Despite this, the LEVEL 1 library was still of great help. Listing of the FEA program is provided in appendix C where all the NAG/SERC library statements appear in a *italic* script whereas the changes to the example chosen are highlighted in **bold** script.

All the information required to model a pressure balance geometry are prepared for a typical 11 runs for each of the piston and cylinder and this will produce 22 data files presenting the "unit" load data characterising the elastic distortion for a balance.

3.11 Implementation of the Computer Models: General

This section describes the purpose built programs which were developed and were required for the modelling of a pressure balance. The section also describes briefly the order in which the programs were run. Some of these programs are based on some libraries and other graphic routines. For example, the FEA program is based on the NAG/FE library while the PVE program uses GINO library subroutines for graphical presentations. Additionally, the FEA program requires that the model geometry is entered prior to it being run. For this purpose, the FEMGEN/FEMVIEW commercial package is used. The modelling of a pressure balance requires, in summary, the following steps:

a. The definition of the pressure balance geometry (using FEMGEN- the pre-processing package).

b. An interface program to adjust the format generated from FEMGEN to match the format required by the NAG/FE library.

c. The characterisation of the elastic distortions of the balance assembly using the purpose built FEA program.

d. Prior to running the PVE program, another interface program is required to adjust the format generated from step **c** (due to the use of the FE library) to match the format required by PVE program. A similar interface program may also be required if it is intended to view the distorted geometry. In the latter case, the format generated due

to the analysis of step **c** has to be matched with the one required by FEMVIEW (the post processing package).

e. The PVE program for the determination of the effective area, distortion coefficient and to display, graphically, the gap and pressure profile along the engagement length of a pressure balance.

All the purpose-built programs for the implementation of steps **b**, **c** and **d** are meant to handle the structural analysis of pressure balances of any geometry. These programs are transparent to the user, therefore no editing or recompilation is necessary when modelling another pressure balance. Likewise is the program in step **e** where the distortion coefficient, pressure and gap profiles can be obtained for any type of lubricating oil or initial clearance gap of a particular pressure balance. The FEA program (step **c**) and the PVE program (step **e**) are the major programs within a total of about 3000 lines of code while the other interface programs are of about 500 lines or so. They are all written in FORTRAN77.

The concept of these programs is described in the following subsections and the introduction of the FE method (chapter 2, section 2.3) which may assist the following up of the listing of the FEA program (Appendix C).

3.11.1 Input Requirements for the FEA Program and its Mechanism

As mentioned earlier, the FEA program will require information on the structure geometry and mesh. This should include the nodes, their coordinates and structure topology. Further information needed would be the material properties (Young's

Modulus and Poisson's ratio) and which nodes are restrained in either x (radial) direction or y (axial) direction (or both). The geometry and topology generated by the pre-processing package (FEMGEN), are saved, for a particular pressure balance geometry, in a file known (conveniently) as **pisina** (**pis**ton-**in**put-**a**nalysis) in the case of the piston and **cylina** (**cyl**inder-**in**put-**a**nalysis) in the case of the cylinder. This is the first requirement. Having created the "pisina" and "cylina" files, their data format is changed to match the requirements of the NAG/SERC (LEVEL 0) format library routines. Appended to that would be the material properties and the restraint information. The later are extracted by viewing the geometry (using FEMGEN) and in the case of the piston, these restrained nodes would be at the top side of the piston, restrained in the y direction (i.e. no allowance for the motion in the y direction) while in the case of the cylinder, they would either be restrained in the y direction only (forming the **sliding support condition**) or some nodes would be restrained (from the motion) in both x (radial) and y direction (forming the **clamped support condition**).

The second input file required for the FEA program is the one containing node numbers at which the elastic distortion may be required (usually along the engagement length). In fact, the user can edit arbitrary node numbers as long as the total number of these nodes is the "header" for this file. It (the file) is known as the **engnodsp** and **engnodsc** for the piston and the cylinder respectively (**eng**agement **nod**es **p**iston and **eng**agement **nod**es **c**ylinder).

The third and final input file required for the FEA program is the assignment of pressure loading on elements of the piston and cylinder. The FEA program can handle any number of loads applied either to individual elements or groups of elements. The input files which will include loading specifications are known as **lodcasp** and **lodcasc** for the piston and cylinder (**load case p**iston and **load case c**ylinder) respectively. For

example, if it is required to apply a pressure load on, say, elements number 1, 2 and 3 simultaneously and then it is intended to apply some other loading on elements 1, 5, 6 and 8 as another load case (and so on), the user can edit these two (or more) load *cases* at once. The **lodcasp** and **lodcasc** files should comply with a certain format and the FEA program will "interpret" these files and it will extract the loaded nodes, their coordinates, calculate and assign the forces (equivalent to the pressure applied in MPa) on each node as well as the total number of loaded nodes. If any mismatch in the advised restricted format is detected, the FEA program will flag an error message and the execution will be aborted. Due to these interpretations, some temporary files are created. For example, for two load cases in the example above, they would appear as "**lcf**001" and "**lcf**002" corresponding to load case file **1** and load case file **2**. Generally, these "lcfxxx" files (where xxx can be from 1 to the total number of load cases) will not be normally displayed to the user but they are accessible, if required, for diagnostic purposes. After obtaining the solution for the elastic distortion at the nodes of interest, all "lcf" files can be deleted.

Having prepared the input files including the geometry, topology, load cases and nodes along the engagement length for a particular pressure balance design, the FEA program would be ready to solve the "unit" load (base) data and produce the output data files containing the engagement length elastic distortions for a pressure balance. For convenience, they are known as **podxxx** and **codxxx** (**p**iston **o**utput **d**istortion and **c**ylinder **o**utput **d**istortion and **xxx** is as in **lcf** above). Normally, these files would be generated starting from the top section to the bottom section along the engagement length. For example, if ten sections are assigned along the engagement length, ten output data files would be generated for each of the piston and the cylinder (**pod1** to **pod10** and **cod1** to **cod10**) respectively. Two more files are usually generated for each of the piston and cylinder (**pod11** in the case of the piston and **cod11** in the case of the

cylinder) and they will correspond to a load case having no load (zero pressure) along the engagement length and the full pressure (1 MPa) away from the engagement length (to incorporate the effect of the applied pressure on the bottom side of the piston). Finally, another two files are generated (known as **pod12** and **cod12** for the piston and the cylinder respectively) and they would include the elastic distortions along the engagement length for the case of a linear pressure distribution applied along the engagement length. This load case would assume 1 MPa away from and at the bottom end of the engagement length and atmospheric pressure ($p = 0$) at the top end of the engagement length. Consequently, the bottom section of the engagement length will have a pressure of 0.95 MPa decreasing to 0.05 MPa at the top section.

The results included in **pod1** to **pod11** and **cod1** to **cod11** are used in the PVE program for the calculation of the effective area, distortion coefficient and displaying graphically the pressure and gap profiles while the results written into **pod12** and **cod12** are used for checking purposes by comparing their values (engagement length radial distortions) with the ones viewed using the FEMVIEW package.

3.11.2 Data Preparation for the PVE Program

The previous section described how the FEA program generates the essential 11 "unit" load data files for the piston (being pod1 to pod11) and for the cylinder (being cod1 to cod11). As the PVE program relies on the application of the superposition method, it is required to prepare each of the 11 data files into one file in the form of a two dimensional array. The program to convert the single dimensional array of each of the 11 data files is known as the **an**alysis **to pve** program (**an2pve**). It basically converts a column vector to a row vector starting from pod1 to pod11 forming what is known as

the **piston** data file **array**, then repeats the same process on the cod1 to cod11 forming the **cylinder** data file **array**. The two latter files are usually named as being related to a particular pressure balance model (e.g. **ps200tun** and **cs200tun** would be the two dimensional array files for the piston and the cylinder for the NPL pressure balance series **200 tun**gsten carbide). These files will be input files to the PVE program for the calculation of the distortion coefficient.

3.12 Application of the Computer Models to the NPL series 400 High Pressure Balance

The detailed application of the models developed are dealt with in the next chapter including the application to NPL and RUSKA pressure balances. In order to validate the models developed, a high pressure balance was chosen as a "test bed" where the oil viscosity effect and the elastic distortions are expected to be significant. The pressure balance chosen was the high pressure NPL (series 400) balance (operating up to 320MPa). The model validation included the generation of the "unit" load data using the FEA program for the piston and cylinder which in turn formed the base data for the PVE program. The following subsections describe the procedure followed in the application of the models.

3.12.1 Support Conditions - Definitions

Prior to introducing the procedure implemented in testing the models, it should be noted that one of the requirements for running the FEA program is to define the support conditions for a pressure balance. This section will define these conditions for the NPL series 400 pressure balance. Other NPL balances are described in chapter 4.

In general, a **sliding support condition** would be if a node or a group of adjacent nodes along an edge is allowed to move either in the x (radial) direction or y (axial) direction. On the other hand, a **clamped support condition** would be if the node(s) is/are **not** allowed to move in either direction. With regard to the piston support conditions, in all the pressure balances modelled, the nodes along the top edge of the piston are allowed to slide radially and are clamped in the y (axial) direction.

In a typical NPL series 400 pressure balance, the piston will float alongside a cylinder in which the shoulder and the base of the cylinder can be defined, as labelled in figure 3.6. The cylinder support conditions can either be:

a. Sliding support where the cylinder shoulder and base are allowed to slide radially.

b. Clamped support where the cylinder shoulder and base are restrained form movement in both x and y directions.

The following section will describe the application of the FEA program to the NPL series 400 pressure balance as a "test bed".

3.12.2 Application of the FEA Model

The FEA can be initiated to generate the "unit" load data characterising the piston and the cylinder. This will form the 22 data files for the piston and the cylinder (11 each) containing the elastic distortions for 1MPa (unit) pressure load. These will be used by the **PVE** program.

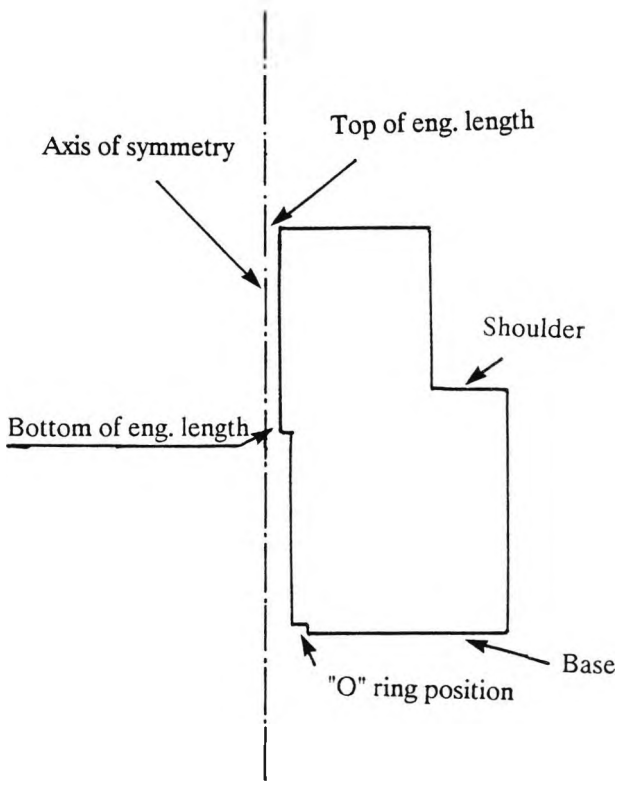


Figure 3.6. Schematic of a cylinder for NPL series 400 pressure balance.

{Note: In the assembly of the balance, a threaded collar (not shown) is hand tightened onto the shoulder of the cylinder which in turn secures the base of the cylinder to its support (also not shown).}

An additional run of the FEA program is to find the elastic distortion for the balance operating at a unit pressure of 1MPa. Results are shown in figure 3.7 for both the cylinder sliding and clamped conditions. To obtain these, the 1MPa pressure was applied at the base of the piston and cylinder along the flanks below the engagement length. Along the engagement length, the pressure was assumed to decrease linearly to zero at its top end. Similarly with the cylinder, a full unit pressure of 1MPa was applied below the engagement length with the exception of the recess at the base where normally an "O" ring would fit. Here the pressure is applied to the vertical face of the recess to simulate the transmission of the radial pressure through the "O" ring when present. With the absence of the "O" ring, both faces of the recess would need to be loaded*. Along the engagement length, a linear distribution of the pressure is applied. In general, a linear distribution is only valid if the initial gap remains constant and pressure-viscosity dependent variations along the engagement length are insignificant.

The "unit" load generation and the 1MPa runs took about two hours on a Sun/SPARC workstation. It can be seen from figure 3.7 that ten sections were assumed along the engagement length. The distorted shape (displayed by the element mesh) is fairly linear for the linear pressure distribution applied. Although the elastic distortion for any pressure can be estimated (details of which are explained in chapter 4, section 4.2.1), for this balance, this will be unrealistic since the viscosity effect is highly significant and the elastic distortion is comparable with the initial gap between the piston and cylinder where the latter is usually small. This is due to the high operating pressure. For the FE analysis of the balance, the material from which the balance was constructed is assumed to be steel (Young's modulus $E = 2.1 \times 10^5$ N/mm^2 and Poisson's ratio of 0.3).

* The "O" ring may transmit a vertical component of force onto the "O" ring recess due to a Poisson effect. However, simulations (using the PVE program), with the presence or the absence of the "O" ring, showed only a 1 % change in the value of the distortion coefficient λ. So, the precise loading condition on the "O" ring recess is not important.

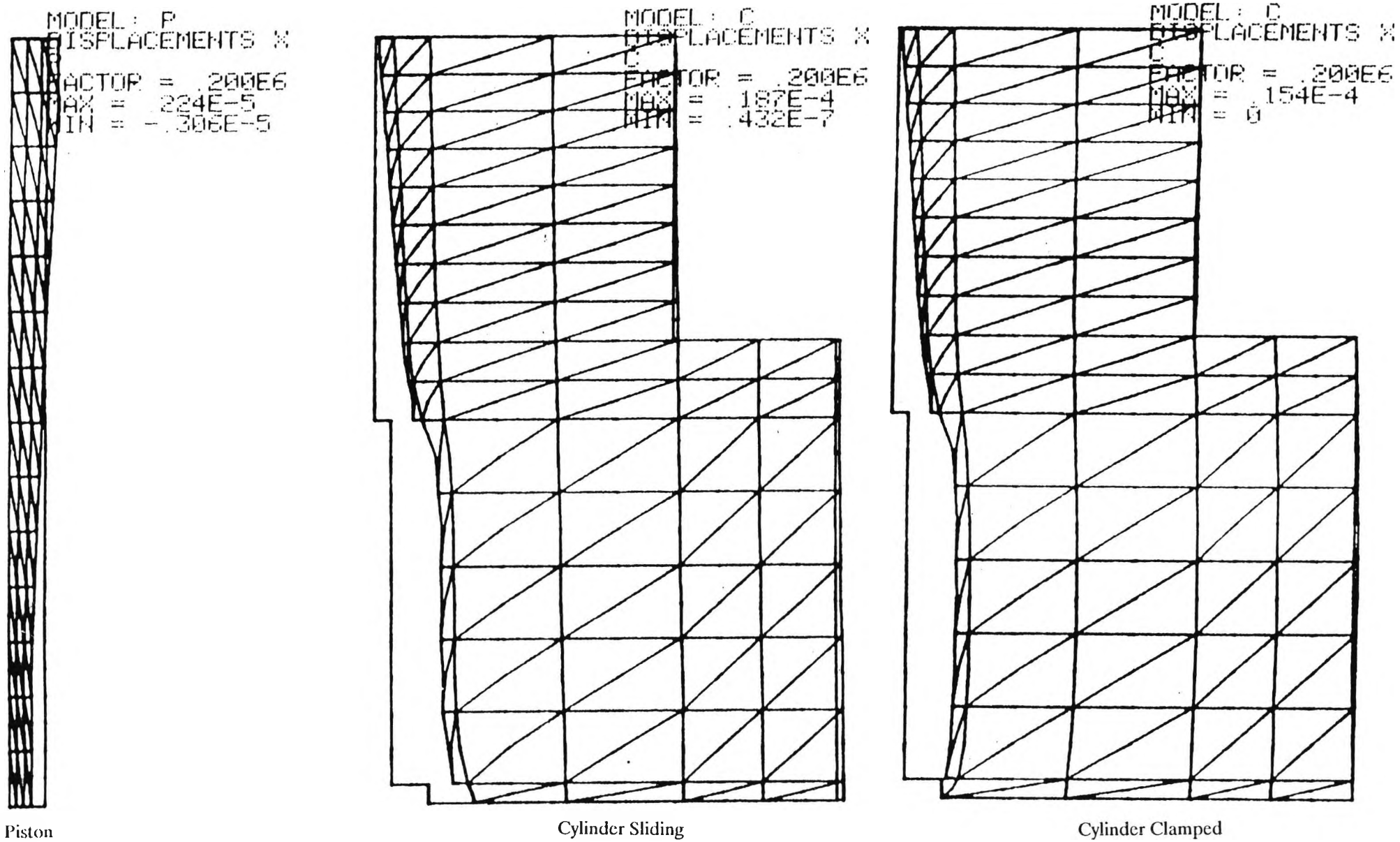


Figure 3.7. Radial distortions at 1 MPa for NPL series 400 pressure balance

In the FEA program, a test run was carried out for 20 sections along the engagement length (figure 3.8 showing the mesh used) and distortion coefficients were then calculated using the **PVE** program. Results were found to be within 1% of those calculated using a 10 section model. Therefore, ten sections have been adopted for all other pressure balances to be simulated in this work. The 20 section FEA model had approximately twice as many elements as the ten section model and a *single* (one load case) run took several hours.

3.12.3 Application of the PVE Model

The "unit" load data were used by the PVE program to calculate the distortion coefficients and present, graphically, the gap and pressure profile for both the piston and cylinder. The distortion coefficients were determined (over the operating range of the balance and up to 320MPa) for the cylinder sliding and clamped support conditions assuming an initial gap of 0.5 μm and a di-2-ethyl hexyl sebacate oil of 21.1 mPa.s viscosity (at ambient pressure) and exponent term (z) of 0.55 (see equation 3.13).

For the sliding or the clamped support, there was virtually no change in the distortion coefficient over the entire operating pressure range (32MPa to 320MPa). The distortion coefficient remained at approximately 3.00 ppm/MPa.

The graphical presentation of the gap and pressure profiles (along the engagement length) at the maximum operating pressure of the balance are shown in figure 3.9. No significant difference was observed for these profiles in the case of cylinder sliding or clamped support conditions. It can be seen that the pressure severely deviates from

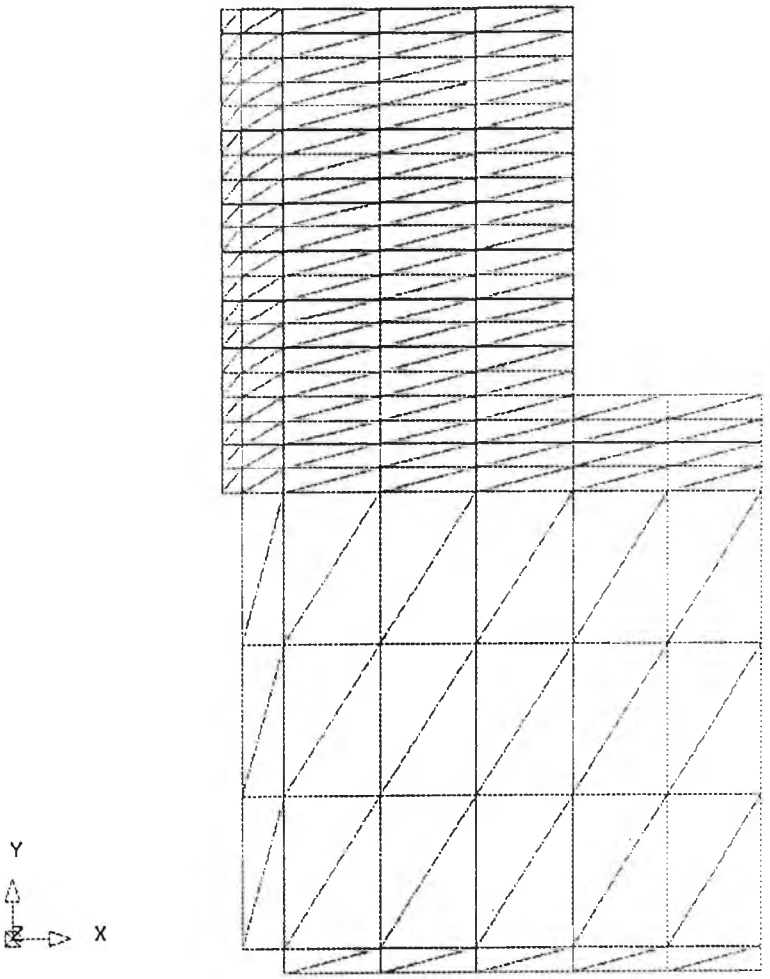


Figure 3.8. The 20 section model for the NPL series 400 pressure balance.

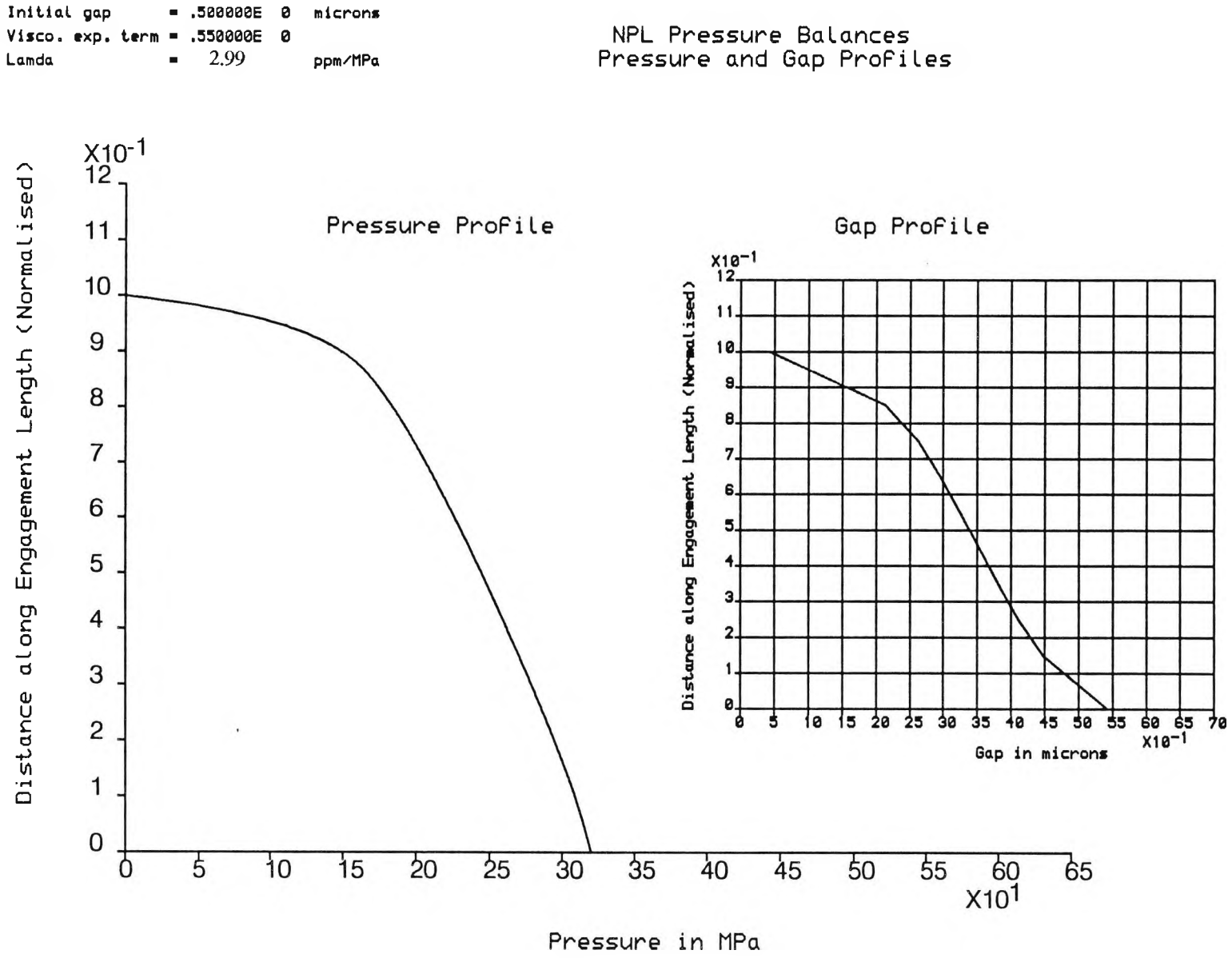


Figure 3.9. Pressure and gap profiles for NPL series 400 balance.

linearity and the major drop is at the top 5-10% of the engagement length. With regard to the gap profile, there is a significant difference in the minimum and maximum gap values. The gap is not constant but it is clearly following the pressure profile. However, the change in the gap is approximately 5.0 μm (in 0.5 μm initial gap) between the top and bottom ends of the engagement length, and the gap tends to close up at the top of the engagement length at very high pressures. For example, a 0.25 μm initial gap produces a negative gap (clearly an impossibility).

To show the effect of the initial gap on the pressure profile along the engagement length and the distortion coefficient, the series 400 pressure balance (sliding support) was simulated for other initial gaps of 1, 2 and 4 μm. It can be seen, from the graphical presentation of these cases in figures 3.10 to 3.12, that there is a wide variation with regard to the pressure profiles (and consequently the gap profiles which follow the pressure ones) but the distortion coefficient remained relatively unchanged with a variation of approximately 7%, (Table 3.2).

NPL Series 400 Summary at 320 MPa Operating Pressure	
Initial Gap (μm)	Distortion Coefficient λ (ppm/MPa)
0.5	2.99
0.5 (and z = 0)	3.05
1.0	3.02
2.0	3.07
4.0	3.20

Table 3.2. Summary of the distortion coefficients for the NPL series 400 balance simulated with different initial gaps.

Another possible investigation would be to simulate the same balance (with a cylinder sliding support condition) with the assumption of having a lubricating oil of no pressure-viscosity dependence. This was simulated by putting the exponent term in the pressure viscosity relationship, z, equal to 0. Results (figure 3.13) show the pressure drop is even more severe than the normal working conditions of the balance as was presented in figure 3.9. The distortion coefficient, however, did not change significantly from the nominal value of approximately 3 ppm/MPa.

Finally, in order to simulate the series 400 balance at a **constant** initial gap, the balance was simulated at an unrealistic initial gap of 32 μm as shown in figure 3.14. It can be seen that the pressure profile obtained from this simulation agrees with a similar result shown by Stuart [20] where the latter analysed high pressure balances incorporating viscosity effects but for a constant initial gaps. With regard to the gap profile, it is evident that the profile is fairly constant while the distortion coefficient was found to be 3.5 ppm/MPa.

For the normal operating conditions of the pressure balance, it may be concluded that the distortion coefficient remained fairly constant regardless of the oil viscosity or the initial gap. This is in good agreement with the conclusion drawn from the analysis reported by Dadson et al [1] where it states that the distortion coefficient is independent of the rheological properties of the transmitting fluid, the piston-cylinder engagement length gap and the applied pressure.

On the other hand, the results obtained are in contrast to those reported by Stuart [20] as the latter work showed that the drop would rather be at the bottom end of the engagement length but the analysis was carried out for a *constant gap*.

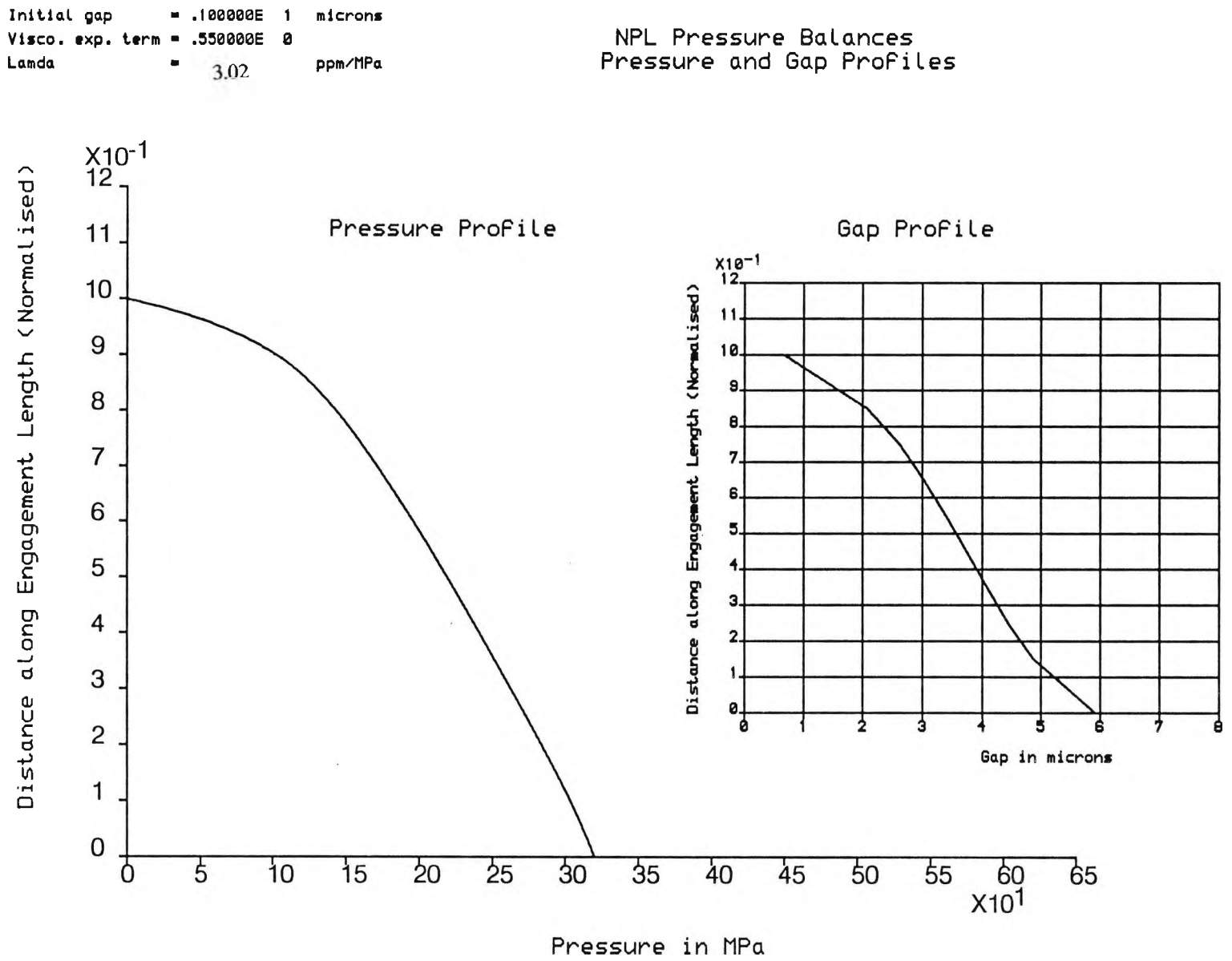


Figure 3.10. Pressure and gap profile for NPL series 400 balance with a 1 μm initial gap.

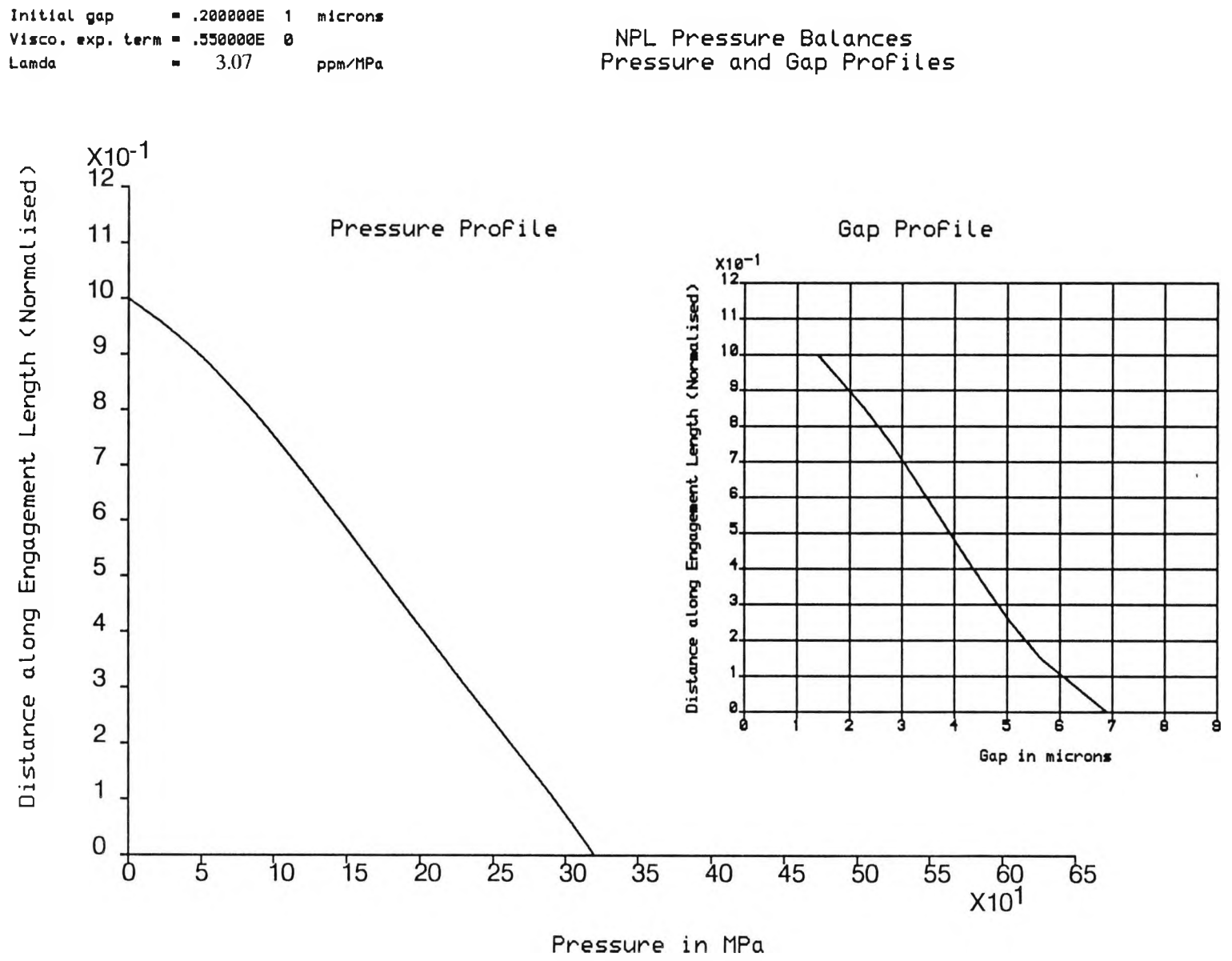


Figure 3.11. Pressure and gap profile for NPL series 400 balance with a 2 μm initial gap.

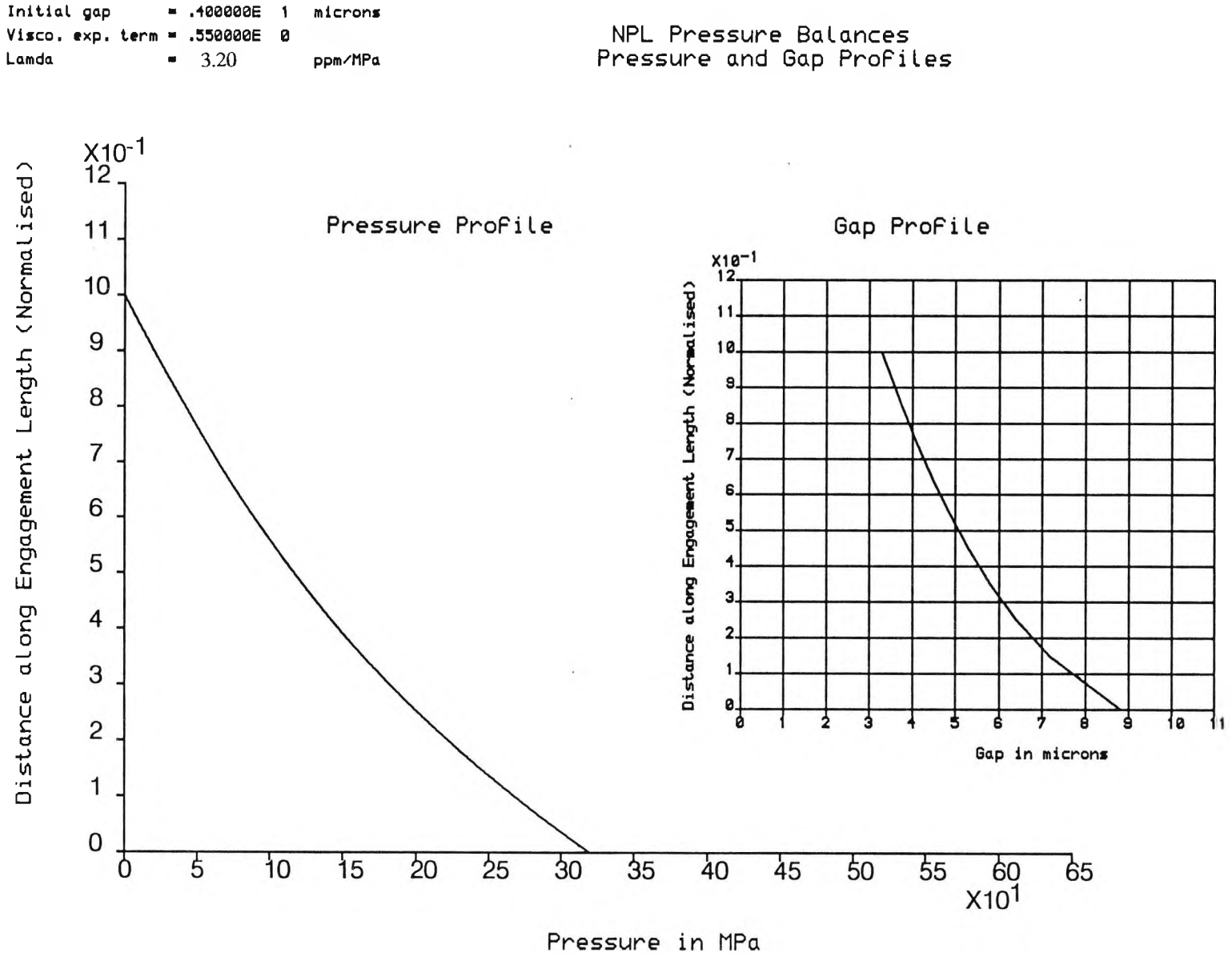


Figure 3.12. Pressure and gap profile for NPL series 400 balance with a 4 μm initial gap.

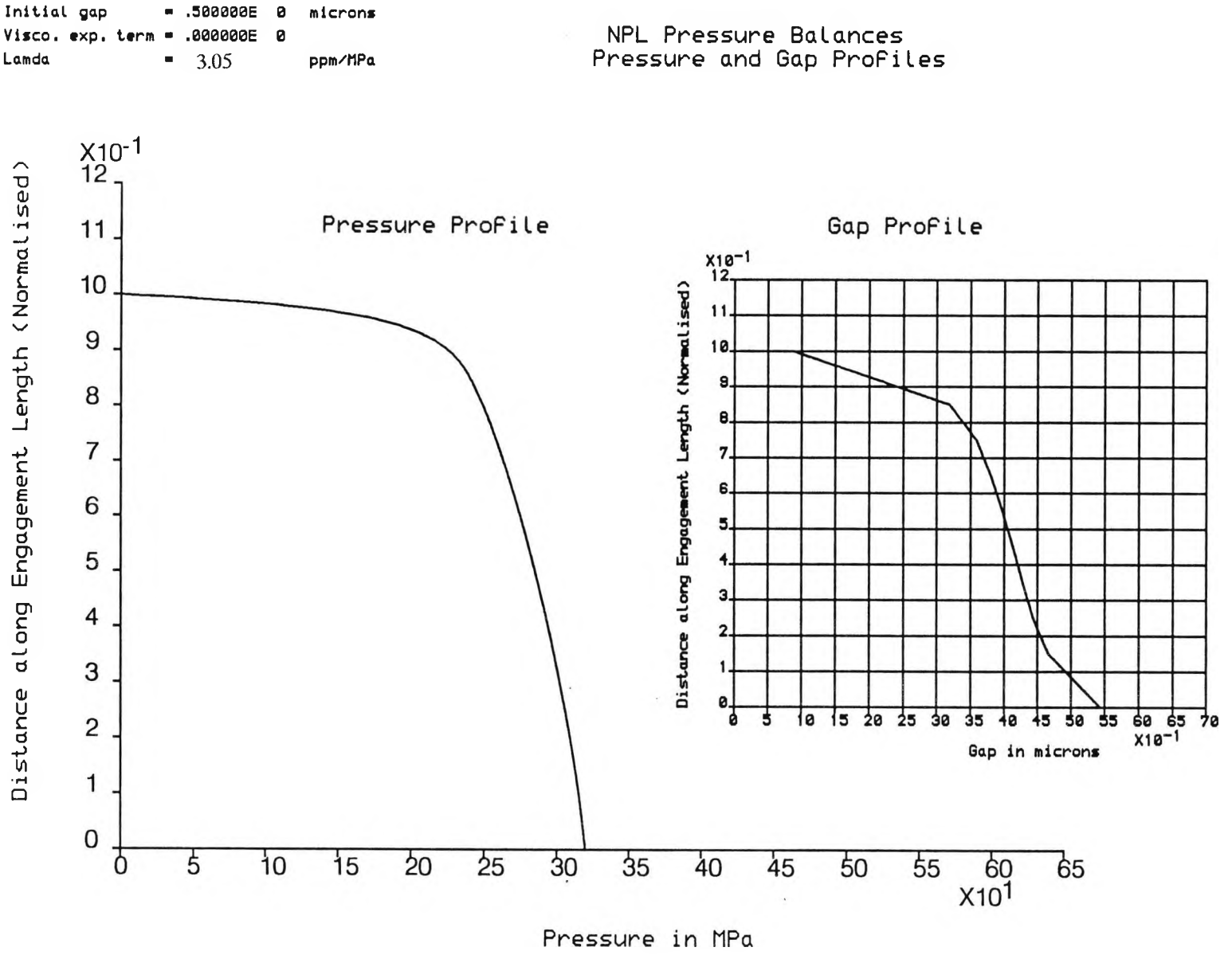


Figure 3.13. Pressure and gap profiles for NPL series 400 balance with oil viscosity exponent term $z = 0$ (therefore viscosity constant).

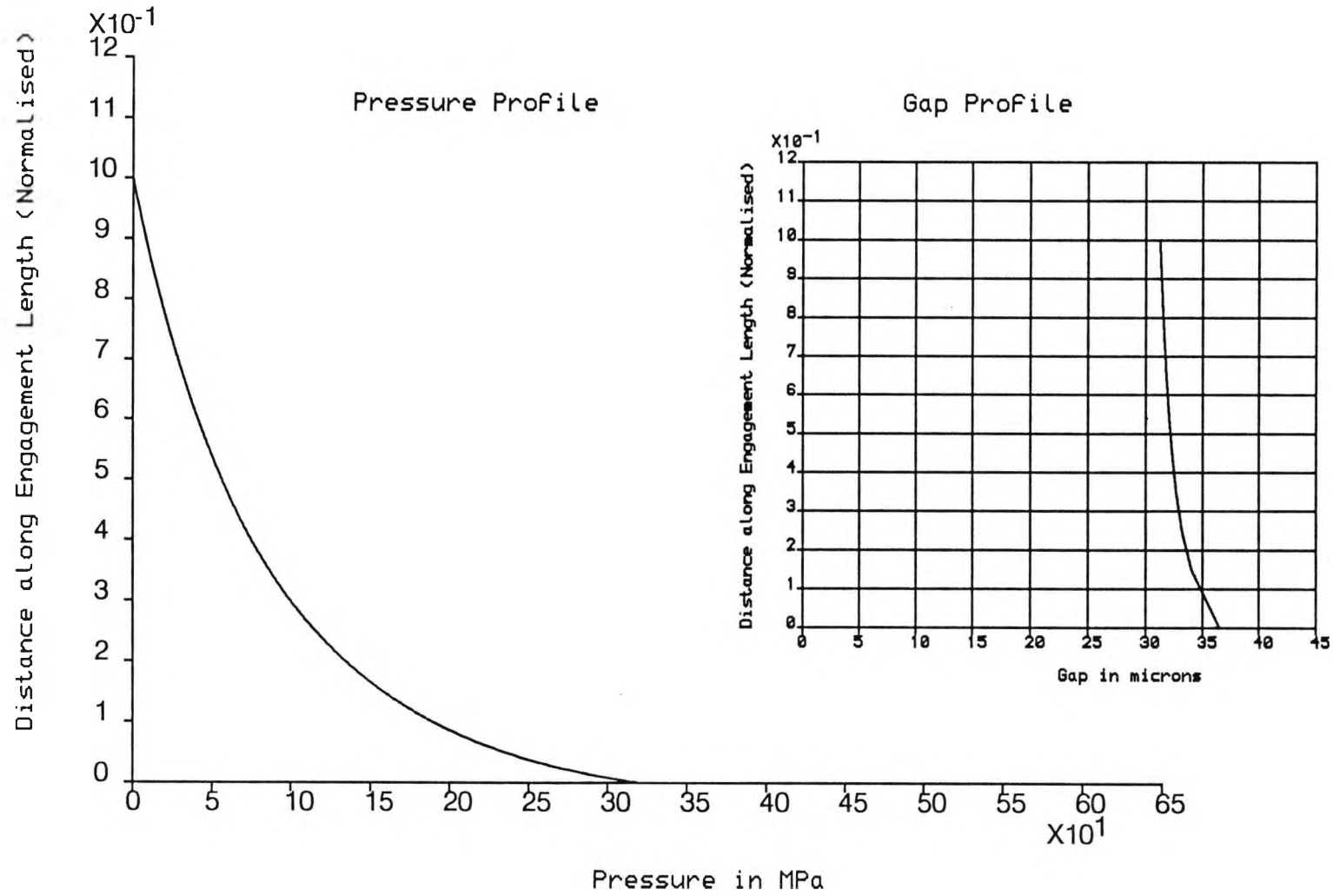


Figure 3.14. Pressure and gap profiles for the NPL series 400 balance with a large initial gap (32 μm) to approximate a constant gap.

3.13 Conclusions

The FEA and the PVE programs have been successfully applied to the NPL series 400 balance with some unexpected conclusions regarding the pressure profile distribution along the engagement length. Simulations for different initial gaps showed a wide variation in the pressure and gap profiles while the distortion coefficient remained unchanged (3.0 ppm/MPa). Further, a simulation for a constant viscosity oil (viscosity exponent term $z = 0$) results in a distortion coefficient of 3.05 ppm/MPa compared to 2.99 ppm/MPa for the real oil. This represents an insignificant change although for the real oil the viscosity changes by a factor of 56 along the engagement length when the balance operates at its maximum pressure of 320 MPa.

The interesting observations on pressure and gap profiles and distortion coefficients presented in the previous section fully justifies the development of the general PVE program. It allows the user to interactively change parameters of any balances. For a given balance, these include, initial gap and viscosity data. A simulation run, including the graphical presentation, takes only a few minutes.

CHAPTER 4

Application to Pressure Balances

4.1 Introduction

The computer models developed for simulating any pressure balance and their testing with the NPL series 400 high pressure balance were described in the previous chapter. In this chapter, the basic methodology developed is applied to other NPL balances and detailed results are presented. Also, the behaviour of the series 400 balance is considered in greater depth since it is the high pressure balance and the pressure-viscosity effect becomes significant. Further, in order to show the generality of the method developed, the simulation of other pressure balances was carried out, namely, the RUSKA 2481 oil operated pressure balance and the RUSKA 2470 gas operated one.

The following section presents the results for the NPL pressure balances while section 4.3 deals with the RUSKA balances.

4.2 The NPL Pressure Balances

The computer models developed were applied to the NPL pressure balances, series 100, 200, 300 and 400. These types cover the primary standards of the NPL pressure range. Series 100 operates up to 5MPa, series 200 up to 20MPa, series 300 up to 80MPa

and finally series 400 pressure balance operating up to 320MPa. A summary of the NPL balances (geometrical, pressure range etc) is shown in Table 4.1a.

It can be seen from Table 4.1a that starting from the lowest pressure range balance (series 100) to the highest one (series 400), it is evident that there is a successive halving of the piston diameter. Also it can be seen that the series 200 to 400 balances, all have the same outer cylinder diameter. For simulating these balances, detailed dimensions are needed and these are provided in Appendix A. Also required are material data (both elastic and fluid) as described in Table 4.1b. Finally, the support conditions for a balance geometry need to be defined. These were briefly stated for the series 400 pressure balance in chapter 3 (section 3.9.1) and are discussed more fully below.

The schematic diagram for the series 100 is shown in figure 4.1a while figure 4.1b shows a typical schematic for the series 200 to 400 pressure balances. In the operation of a balance, the method of loading the top of the piston with weights is such that there is likely to be negligible restraints in the x (radial) direction so the boundary condition applied at the top of the piston is a restraint in the y (axial) direction only. Therefore the piston will always posses the sliding support condition. As regard to the cylinder, it is assumed that the shoulder and base (see figure 4.1b) will slide radially so these two edges are restrained in the y direction. This is defined as the cylinder **sliding** support **condition** as it is expected to be the normal operating condition. However, it is possible that the clamping collar (not shown) may be tightened too firmly when assembling the balance leading to the prevention of sliding at the cylinder shoulder and/or base. The worst condition would be if both shoulder and base are restraint from movement in both x (radial) and y (axial) directions. This is defined as the cylinder **clamped** support **condition.**

Geometrical Data	NPL Simple Geometry Balances Series			
	100	200	300	400
Type of Balance	Oil			
Oil used	Sebacate			
Press. Range (MPa)	5.0	20	80	320
Eng. length (mm)	74.0	40	35	25
Piston Radius (mm)	12.5	6.25	3.125	1.5625
Cylinder OD (mm)	32.5	23.75	26.875	28.45

(a)

Symbol	Description	Unit	Value
E	Youngs modulus	N/mm^2	$2.10x10^5$
ν	Poissons ratio		0.30
η_o	Visc. at amb. press.	mPa.s	21.1
z	Exp. term (eq. 3.13)		0.55

(b)

Table 4.1 Summary of (a) the geometrical and (b) the fluid data for the NPL series of balances

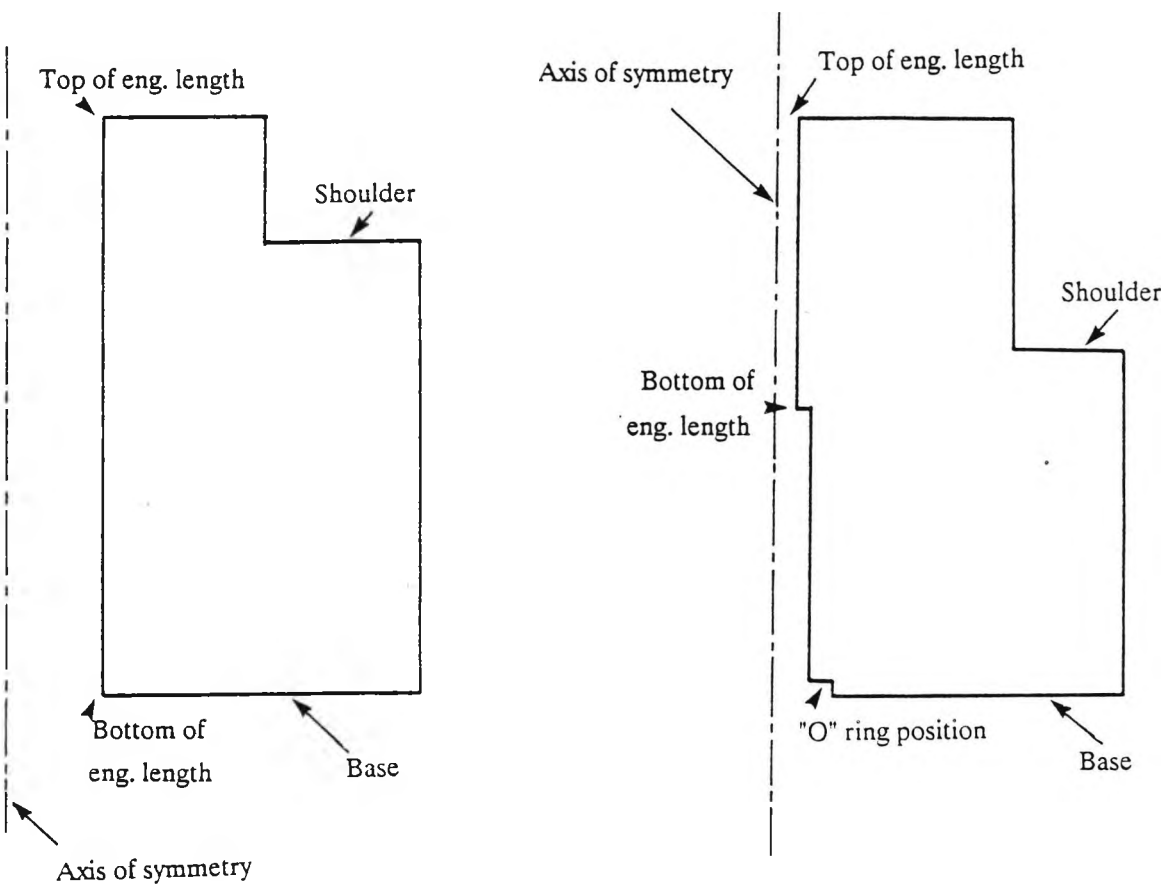


Figure 4.1. Schematic of cylinders for the NPL pressure balances.
(a) Series 100. (b) Series 200, 300 and 400.

The simulation carried out for these type of balances was essentially following the same procedure as for the NPL series 400 pressure balances described in chapter 3 (section 3.9). That is, for each balance, and using the FEA program, the preparation of the "unit" (base) data is performed so obtaining the elastic distortions at unit pressure loading of 1MPa for 10 sections along the engagement length for both the piston and cylinder. Having obtained the "unit" load data, the PVE program was run to determine and present graphically the pressure and gap profiles along the engagement length and calculate the distortion coefficient of an assembly. In addition, for each balance, the piston and cylinder elastic radial distortions at 1 MPa operating pressure are presented. These can be used to estimate distortions at any given pressure by a proportional scaling of the 1MPa results provided that the gap remains essentially constant and viscosity variations are small. Then the pressure distribution along the engagement length remains a linear one, so scaling is possible. The validity of this scaling is described for each balance in the following subsections.

Further, for all NPL balances, the simulation assumes a gauge mode of operation where the top end of the engagement length experiences the atmospheric pressure.

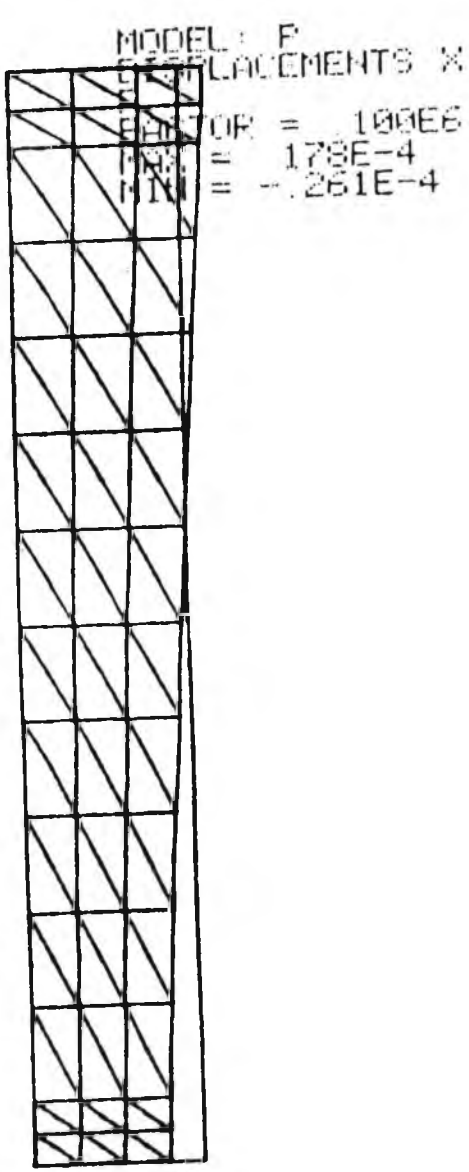
4.2.1 The NPL Series 100 Pressure Balance

Using the FEA model, the radial distortions for 1MPa are shown in figure 4.2 with a magnification factor of 10^6 and it can be seen that the distorted shape (displayed by the element mesh) is almost linear. One can estimate proportionally, the actual elastic distortion for any operating pressure from these results. In order to do that, the dimensions of the piston and cylinder have to be known. For example, for the cylinder sliding support, the movement (or distortion) at the bottom of the cylinder (which is

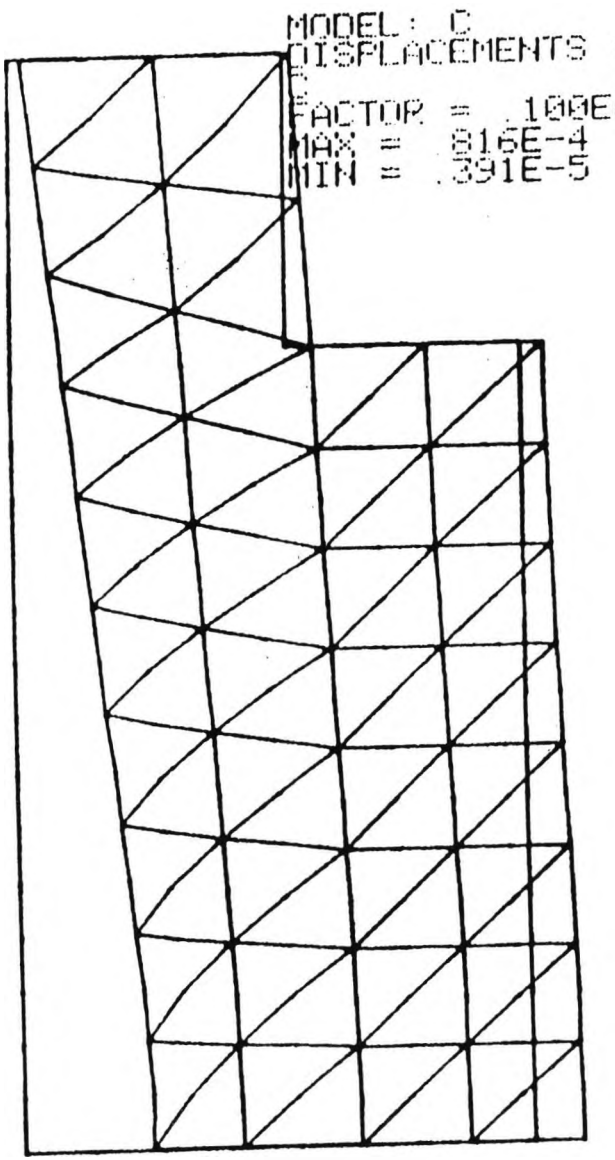
the bottom end of the engagement length) can be estimated to be 0.25 of the base length (32.5mm - see Appendix A) resulting in a movement of approximately 0.08 μm. Then at 5 MPa, this movement will be 0.4 μm. This is the largest movement in the balance but is small compared to the initial gap of 1.5 μm. Since this balance is a low pressure one, and as the pressure profile is shown to be almost linear (see below) meaning that the viscosity effect can be ignored, then this scaled estimation of the elastic distortions will not be a bad approximation.

The graphical presentation (using the **PVE** program) of the gap and pressure profiles along the engagement length, at the maximum operating pressure of the balance, is shown in figures 4.3 and 4.4 for the cylinder sliding and clamped conditions respectively. It can be seen that in both cases, the pressure varies linearly between the top and bottom end of the engagement length. However, some difference in the gap profile can be seen if one compares the sliding and clamped support conditions. In both cases, the gap is not constant although there is a change of approximately 0.5 μm (in 1.5 μm initial gap) between the top and bottom ends of the engagement length.

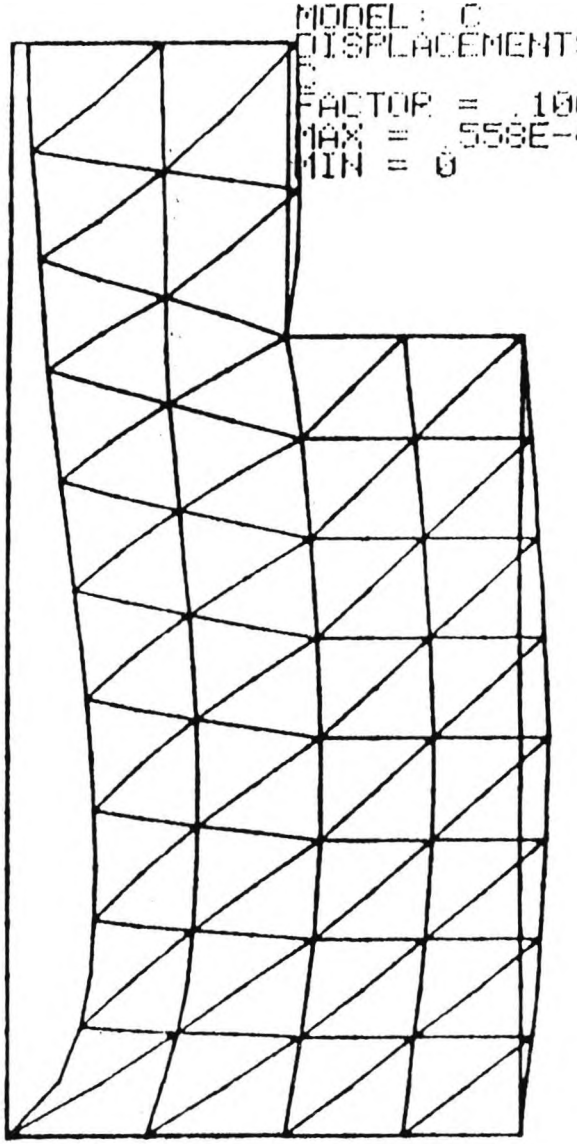
Using the same (PVE) program, for the sliding support, it was found that the distortion coefficient varies between 3.25 ppm/MPa at an operating pressure of 0.5 MPa to 3.29 ppm/MPa at the maximum operating range of 5.0 MPa. As regard to the clamped support, the distortion coefficient varied between 2.28 ppm/MPa to 2.33 ppm/MPa for the same operating pressure range. The decrease in the distortion coefficient is approximately 30% between the sliding and clamped support conditions. The results of the PVE program are presented (over the entire pressure range) for the both support conditions in Table 4.2. This result is in good agreement with the work reported by Dadson et al [1] and Peggs et al [15] since at low pressures, which is the case in this balance, there will be no viscosity effect.



Piston



Cylinder Sliding



Cylinder Clamped

Figure 4.2. Radial distortions at 1MPa for NPL series 100 pressure balance (magnification 1 x 10^5).

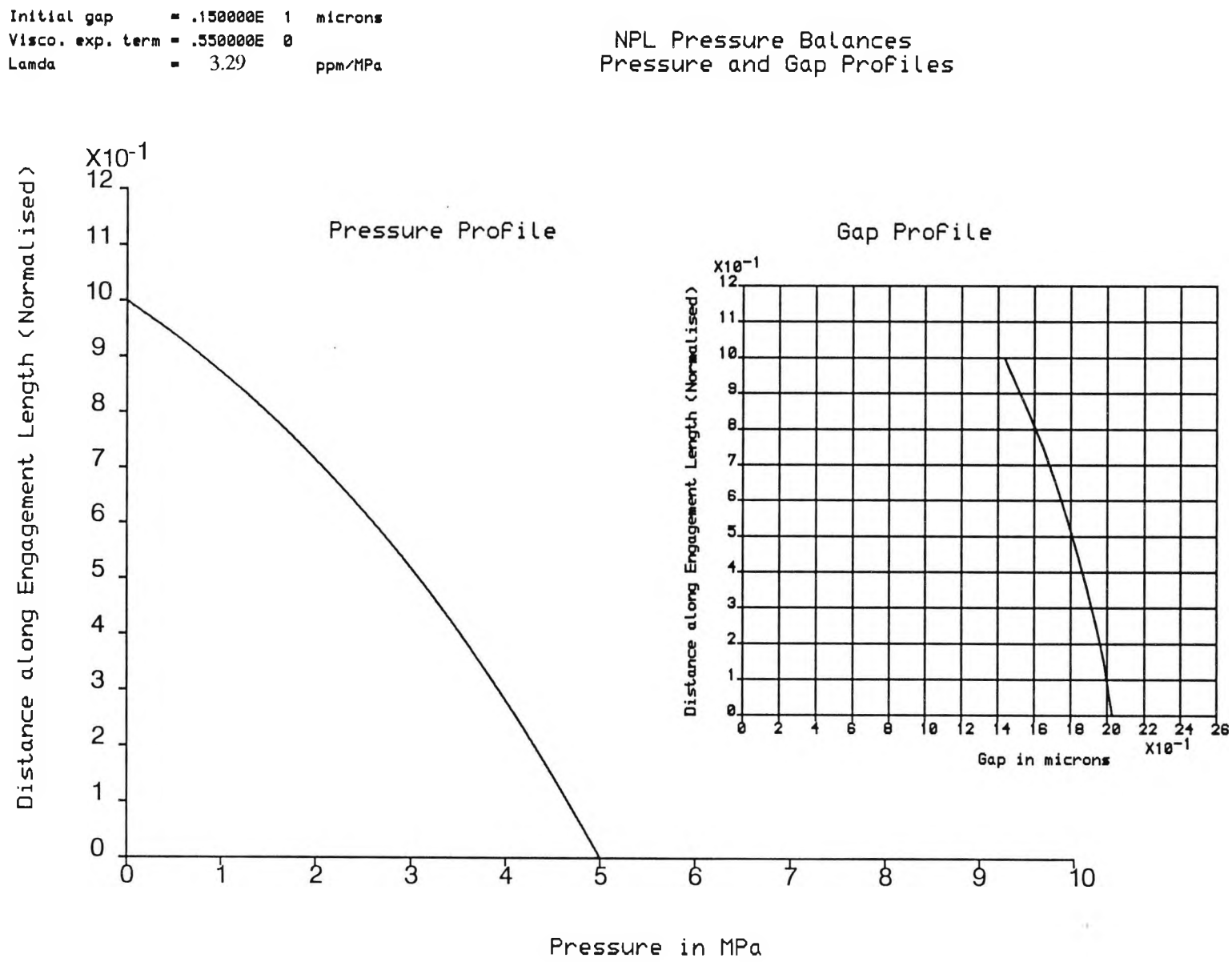


Figure 4.3. Pressure and gap profiles for NPL series 100 balance (cylinder - sliding support).

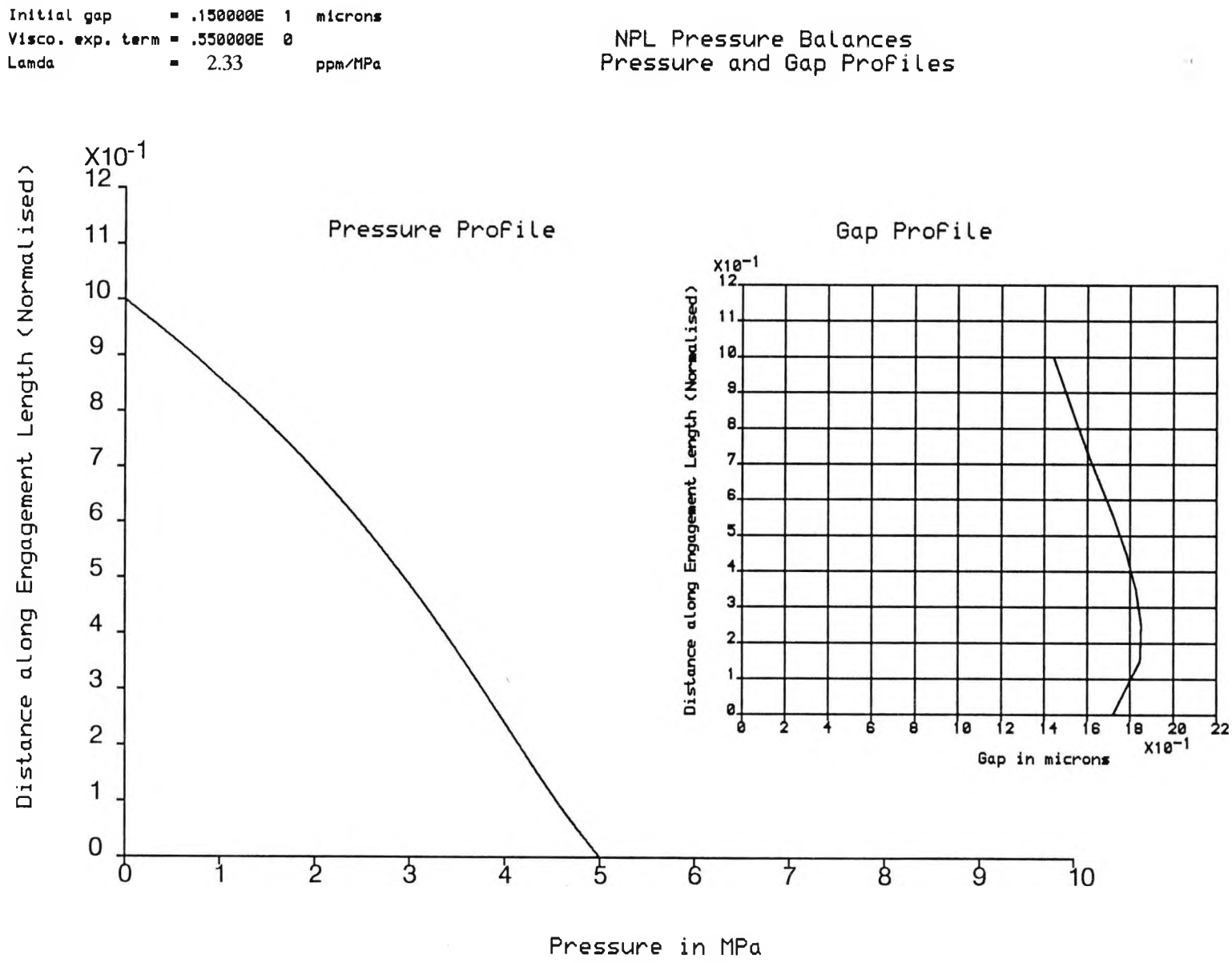


Figure 4.4. Pressure and gap profiles for NPL series 100 balance (cylinder - clamped support).

Series 100, Initial gap 1.5 μm, Nominal area 490.9327 sq.mm.		
Pressure (MPa)	Distortion Coefficient λ (ppm/MPa)	
	Sliding	Clamped
0.50	3.25	2.28
1.00	3.26	2.28
1.50	3.26	2.29
2.00	3.27	2.29
2.50	3.27	2.30
3.00	3.27	2.31
3.50	3.28	2.31
4.00	3.28	2.32
4.50	3.29	2.32
5.00	3.29	2.33

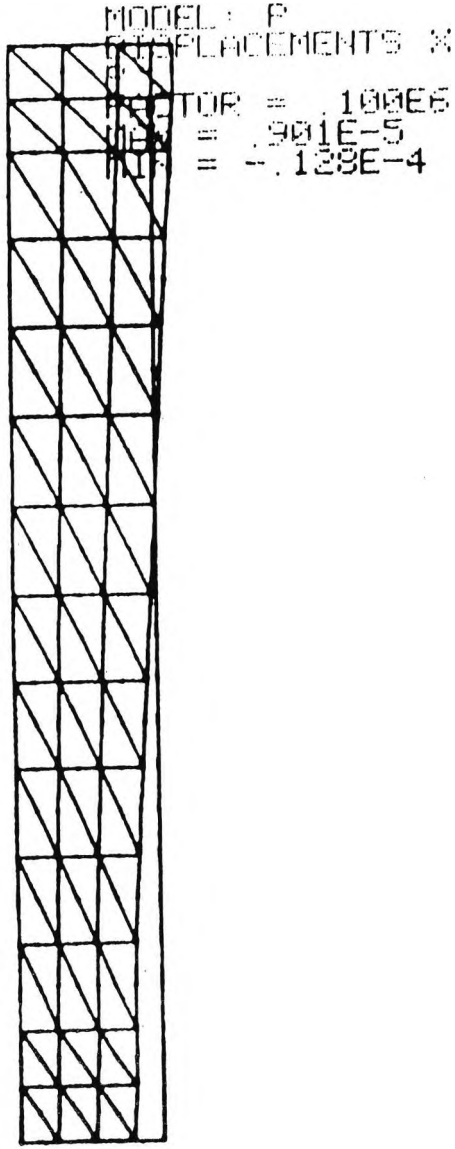
Table 4.2. Distortion coefficients for the NPL series 100 pressure balance

4.2.2 The NPL Series 200 Pressure Balance

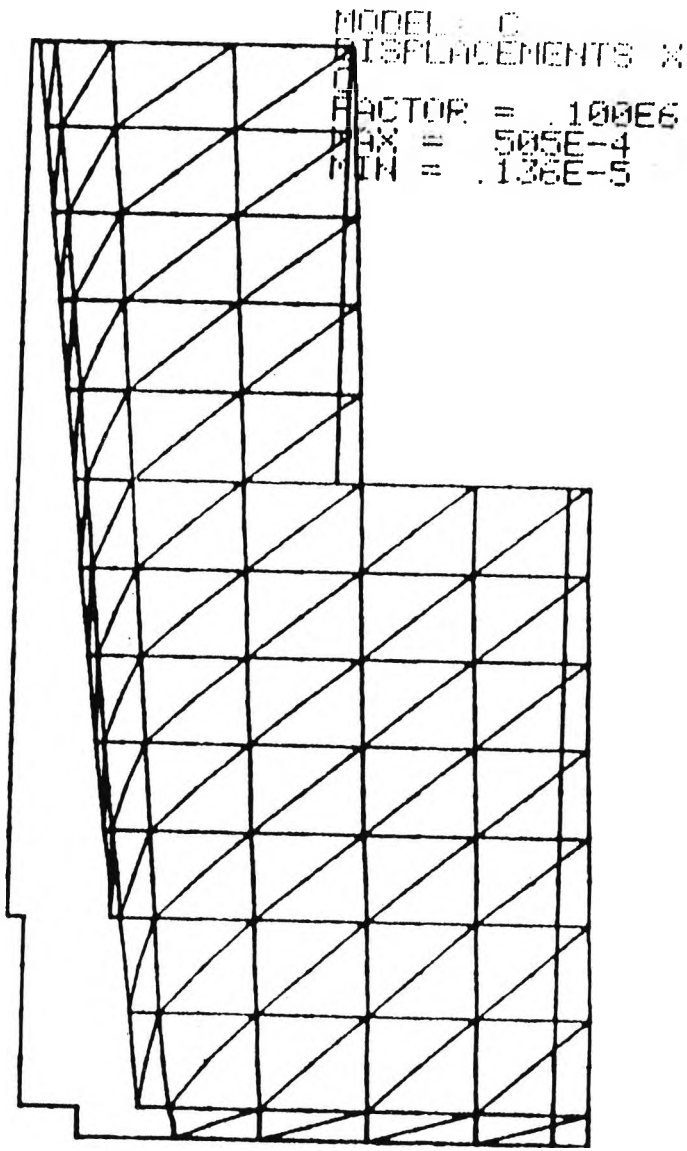
The radial distortions for 1MPa are shown in figure 4.5. The distorted shape (displayed by the element mesh and obtained using the FEA program) is fairly linear for the 1MPa applied pressure (with an assumed linear pressure distribution along the engagement length as for the series 100). One can estimate the elastic distortion for any pressure. Again, for this balance, this will not be a bad estimation as this balance still operates at relatively low pressures which implies that the elastic distortion are somewhat small compared to the initial gap of 1.0 μm and the effect of the viscosity variations can be ignored as is confirmed by the relatively linear pressure profile along the engagement length (see below).

As for the series 100 balance, the pressure and gap profiles are presented and the distortion coefficients are determined (using the PVE program) for both the cylinder sliding and clamped support conditions. The pressure and gap profiles along the engagement length at the maximum operating pressure of the balance, for the sliding support, is shown in figure 4.6 while figure 4.7 shows the same profiles but for the cylinder clamped conditions. It can be seen that in both cases, the pressure deviates somewhat from linearity compared to series 100 pressure balance. As regard to the gap profile, there is some difference in the minimum and maximum gap for the sliding and clamped support conditions. Again, in both cases, the gap is not constant and is clearly following the pressure profile. There is a change of approximately 1.1 μm (in 1.0 μm initial gap) between the top and bottom ends of the engagement length.

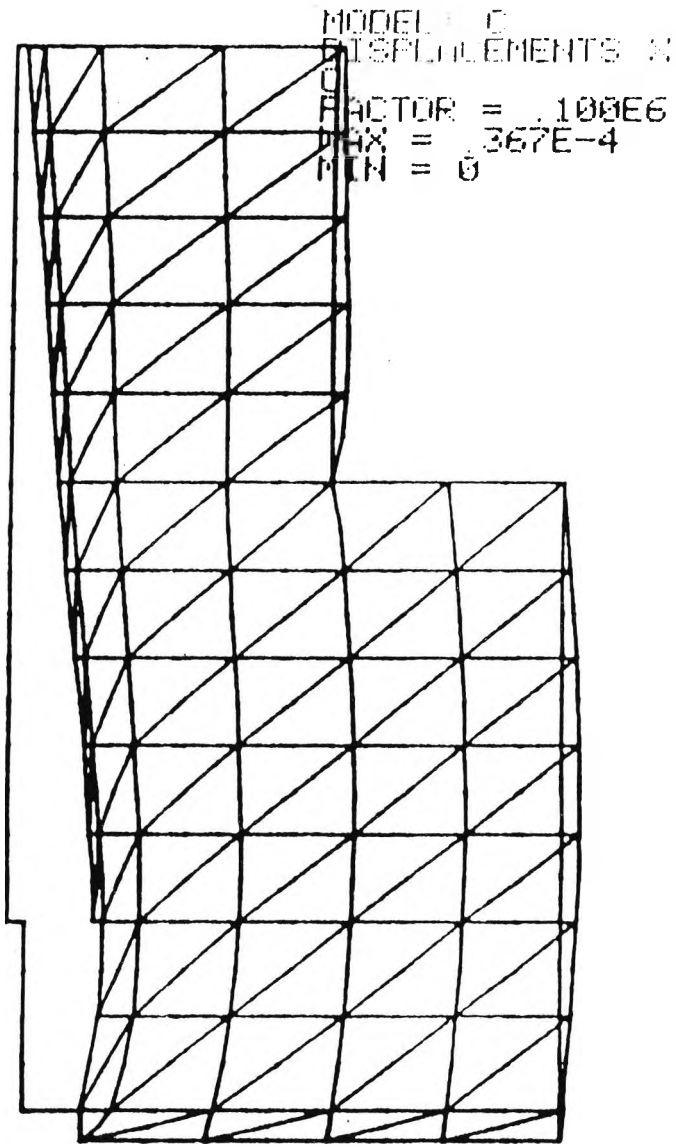
As regard to the distortion coefficients, for the sliding support, it was found that λ varies between 3.27 ppm/MPa at an operating pressure of 2.00MPa to 3.32 ppm/MPa at the



Piston



Cylinder Sliding



Cylinder Clamped

Figure 4.5. Radial distortions at 1 MPa for NPL series 200 pressure balance (magnification 1 x 10^5).

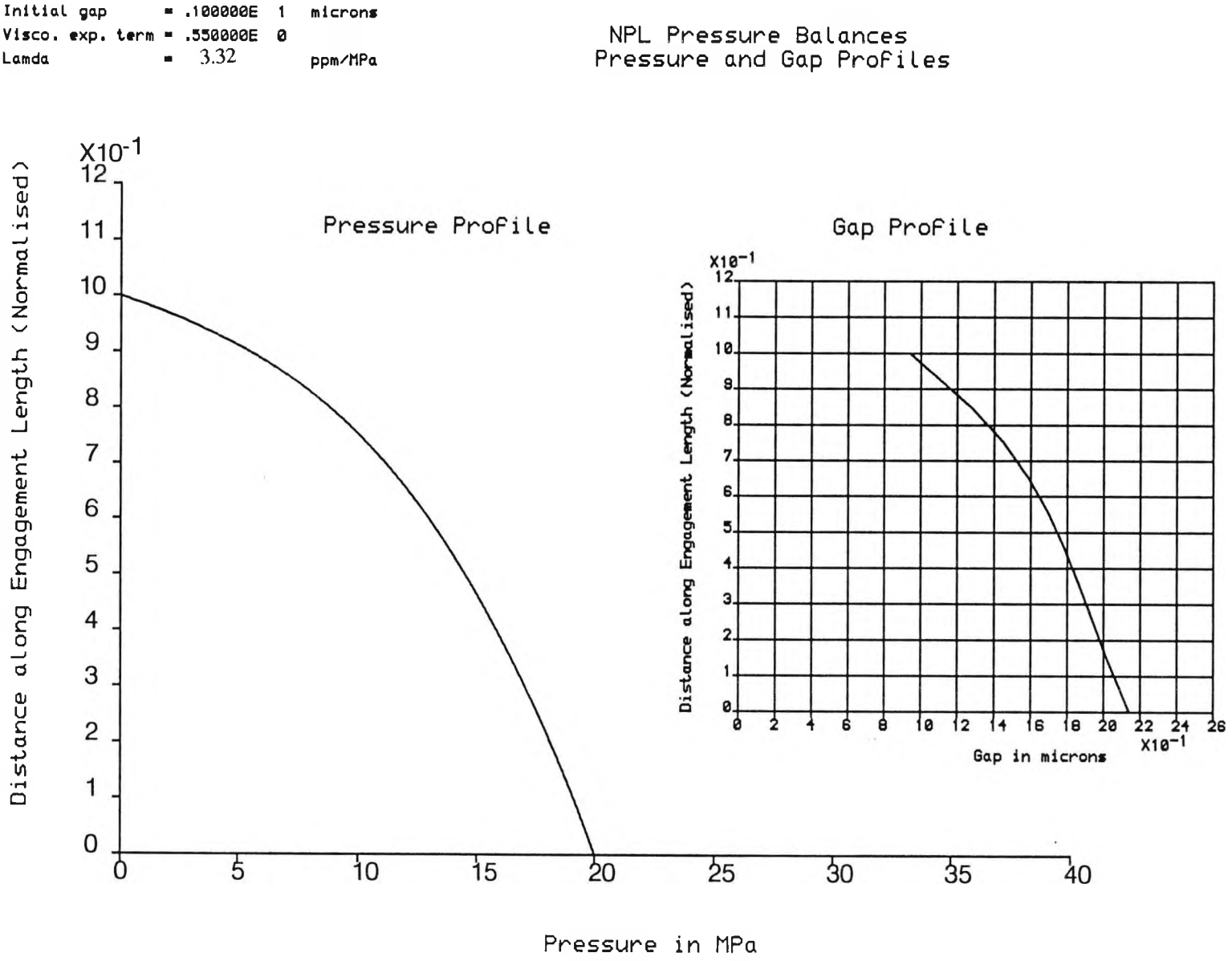


Figure 4.6. Pressure and gap profiles for NPL series 200 balance (cylinder - sliding support).

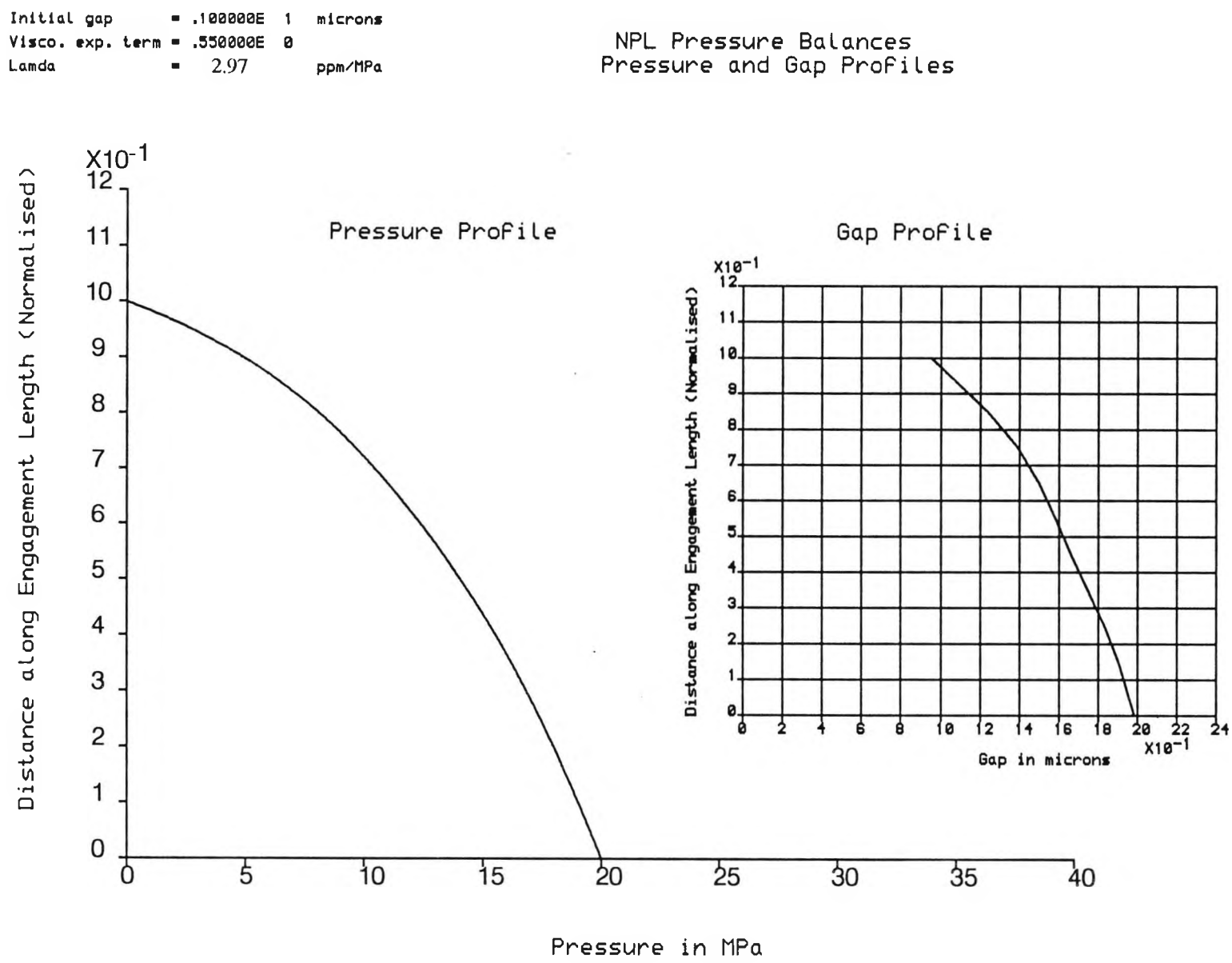


Figure 4.7. Pressure and gap profiles for NPL series 200 balance (cylinder - clamped support).

Series 200, Initial gap 1.0 μm Nominal area 122.7381 sq.mm.		
Pressure (MPa)	Distortion Coefficient λ (ppm/MPa)	
	Sliding	Clamped
2	3.27	2.83
4	3.27	2.84
6	3.27	2.89
8	3.28	2.87
10	3.28	2.89
12	3.29	2.91
14	3.30	2.92
16	3.31	2.94
18	3.31	2.96
20	3.32	2.97

Table 4.3. Distortion coefficients for the NPL series 200 pressure balance

maximum operating range of 20MPa. As regard to the clamped support, the distortion coefficient varied between 2.83 ppm/MPa to 2.97 ppm/MPa for the same operating pressure range. The decrease in the distortion coefficient is approximately 12% between the sliding and clamped support conditions. The results of the PVE program are presented (over the entire pressure range) for the sliding and clamped support conditions in Table 4.3.

Finally, the series 200 pressure balance was simulated as being made of tungsten carbide. The distortion coefficient was found to be constant at a value of 0.94 ppm/MPa over it operating pressure range. This value is to be compare with a theoretical value of 0.87 ppm/MPa obtained from equation 104 of Dadson et al [1] and an experimental value of 1.5 ppm/MPa provided by NPL for this balance [26].

4.2.3 The NPL Series 300 Pressure Balance

For the series 300 balance, the radial distortions for 1MPa are shown in figure 4.8. The distorted shape (displayed by the element mesh and obtained using the FEA program) is fairly linear for the 1 MPa pressure distribution applied. For this balance, estimating elastic distortions at high pressures by scaling the 1MPa results (of figure 4.8) will not be as a good an approximation as the estimations for the series 100 and 200 balances. This is because the pressure becomes high enough so that the elastic distortion are comparable with the 1.0 μm initial gap. In addition, the effect of the dynamic viscosity becomes potentially significant. The resulting pressure profile becomes far from a simple linear distribution necessary for scaling.

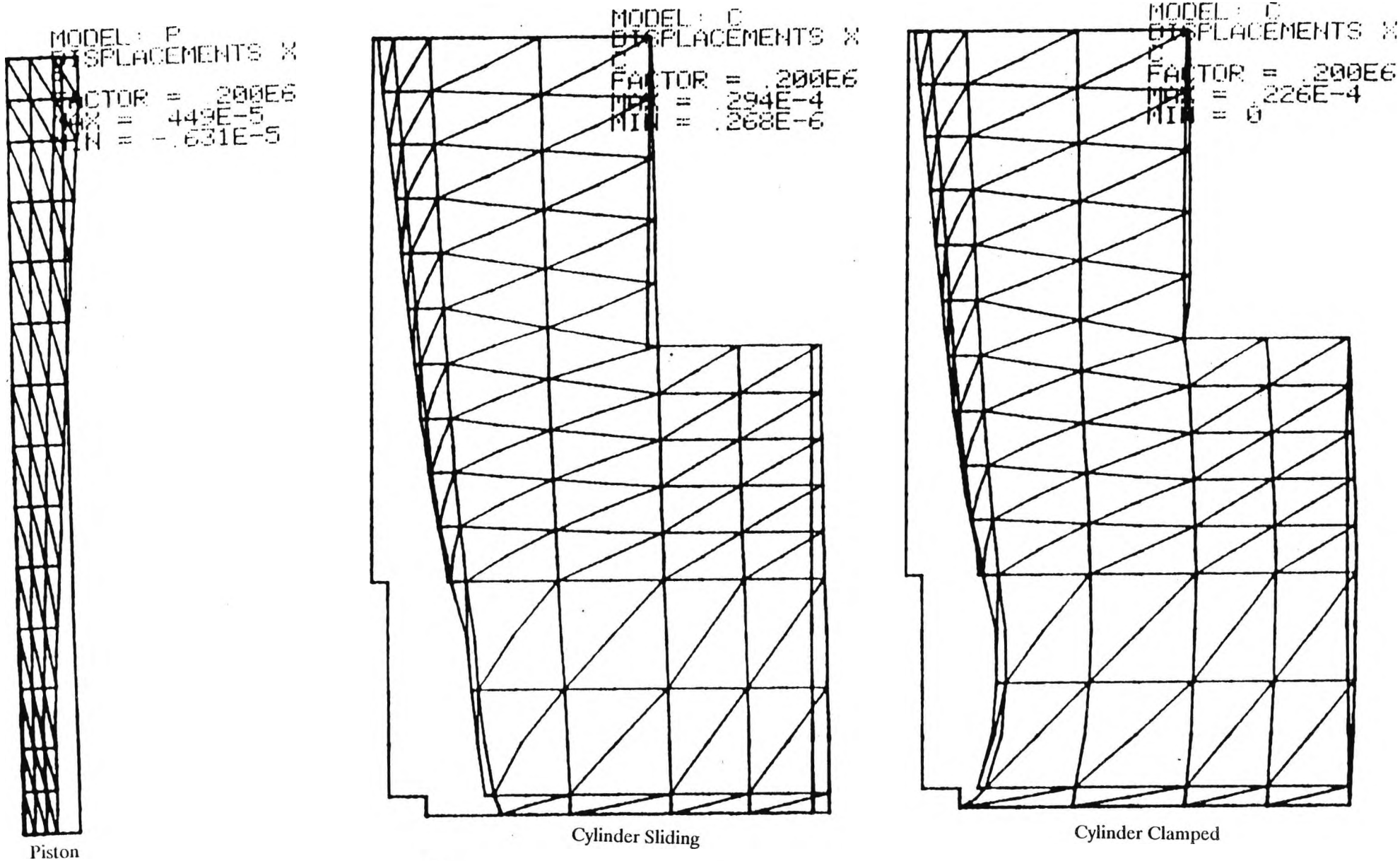


Figure 4.8. Radial distortion at 1 MPa for NPL series 300 pressure balances (magnification 2 x 10^5).

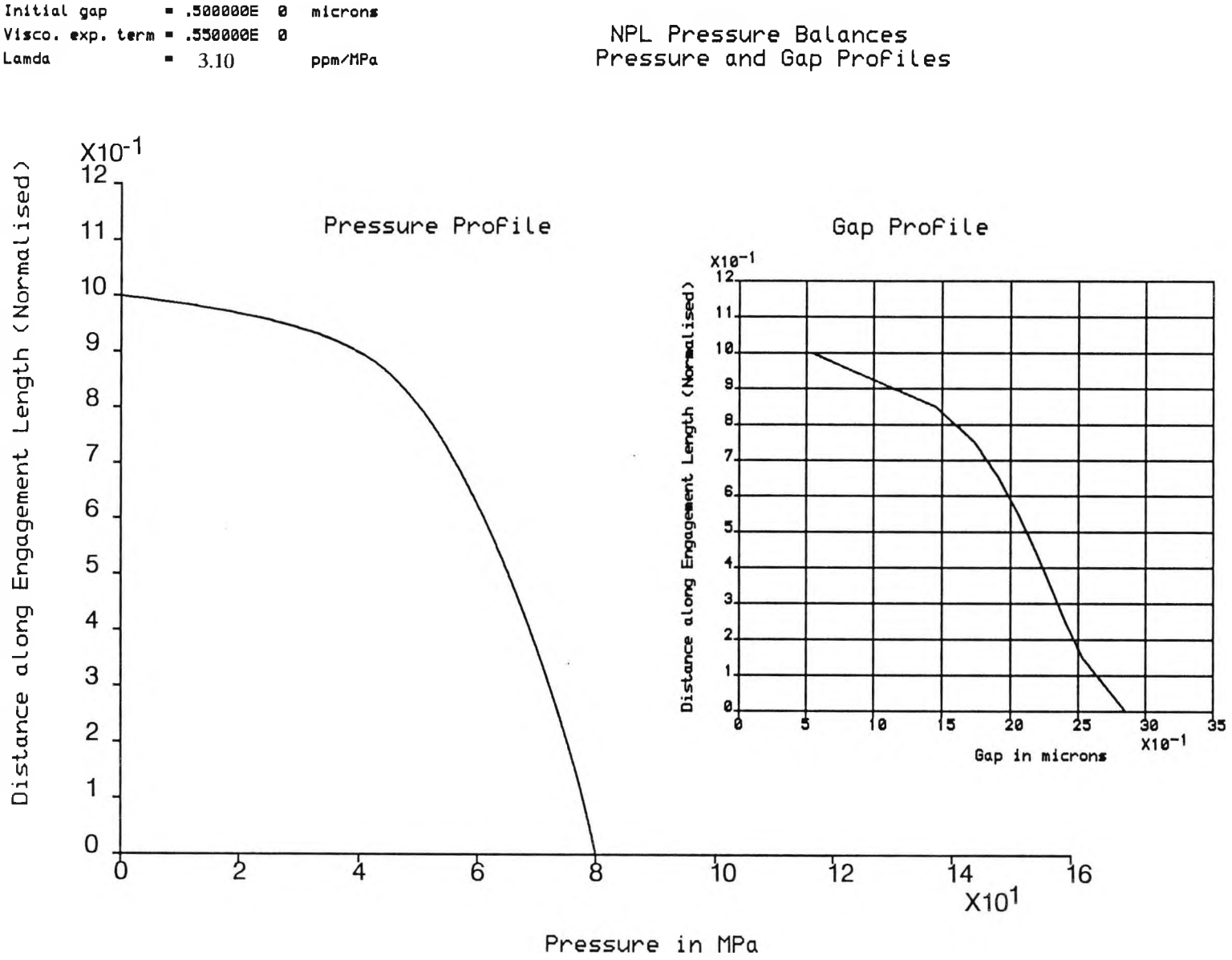


Figure 4.9. Pressure and gap profiles for NPL series 300 balance (cylinder - sliding support).

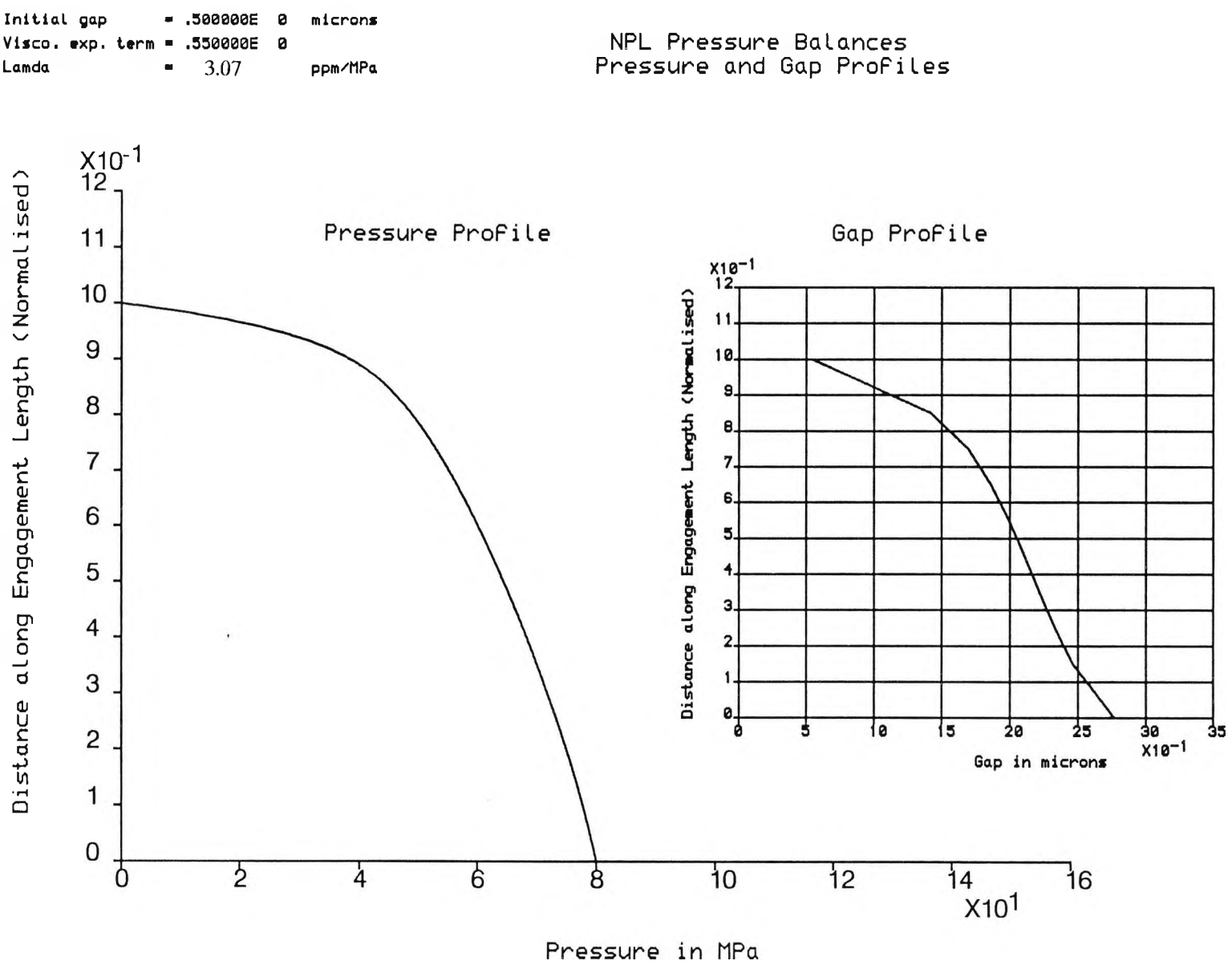


Figure 4.10. Pressure and gap profiles for NPL series 300 balance (cylinder - clamped support).

Series 300, Initial gap 0.5 μm Nominal area 30.68452 sq.mm.		
Pressure	Distortion Coefficient λ (ppm/MPa)	
(MPa)	Sliding	Clamped
8	3.04	2.96
16	3.03	2.97
24	3.04	2.98
32	3.05	3.01
40	3.06	3.02
48	3.07	3.04
56	3.08	3.05
64	3.09	3.06
72	3.09	3.07
80	3.10	3.07

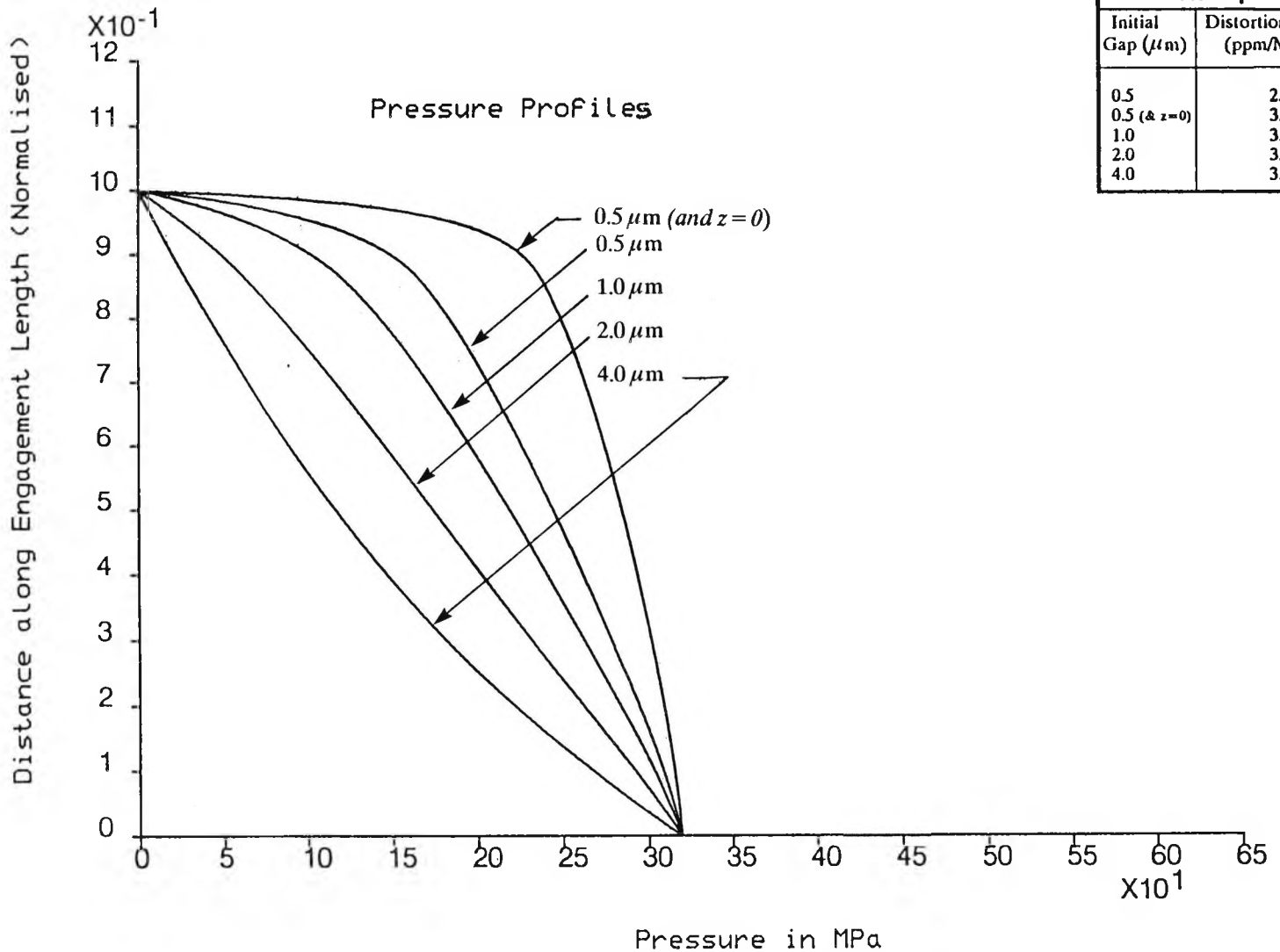
Table 4.4. Distortion coefficients for the NPL series 300 pressure balance

The gap and pressure profiles along the engagement length at the maximum operating pressure of the balance, for the sliding support, are shown in figure 4.9 while figure 4.10 shows the same profiles but for the cylinder clamped conditions. It can be seen that in both cases, the pressure deviates significantly from linearity compared to series 100 or series 200 pressure balance. As regard to the gap profile, there is a significant difference in the minimum and maximum gap for the sliding and clamped support conditions. Again, in both cases, the gap is not constant and is clearly following the pressure profile. The change in the gap is approximately 2.2 μm (in 0.5 μm initial gap) between the top and bottom ends of the engagement length.

The distortion coefficients were also determined using the PVE program and for the sliding support, it was found that the λ varies between 3.04 ppm/MPa at an operating pressure of 8.00MPa to 3.10 ppm/MPa at the maximum operating range of 80MPa. As regard to the clamped support, the distortion coefficient varied between 2.96 ppm/MPa to 3.07 ppm/MPa for the same operating pressure range. For this balance, there is essentially little change in the distortion coefficient between the two support conditions. The results of the PVE program are presented (over the entire pressure range) for the sliding and clamped support conditions in Table 4.4.

4.2.4 The NPL Series 400 Pressure Balance operating up to 320MPa

In the previous chapter (section 3.9), the NPL series 400 balance was chosen as a test bed to show the validity of the models. Presented there were the results for this balance using both the FEA and PVE programs. It could be seen that pressure profile changes very significantly between a simulation of the balance with a constant gap compared to the simulation with a converged actual gap. This confirms the significant effect of



Series 400 Summary at 320 MPa Operating Pressure	
Initial Gap (μm)	Distortion Coefficient λ (ppm/MPa)
0.5	2.99
0.5 (& z=0)	3.05
1.0	3.02
2.0	3.07
4.0	3.20

Figure 4.11. Pressure profiles for the NPL series 400 pressure balance simulated with different initial gaps.

Series 400, Initial gap 1.0 μm Nominal area 7.674814 sq.mm.		
Pressure	Distortion Coefficient λ (ppm/MPa)	
(MPa)	Sliding	Clamped
32	3.03	3.00
64	3.01	2.99
96	3.00	2.99
128	3.00	2.99
160	3.00	2.99
192	3.00	2.99
224	3.01	3.00
256	3.01	3.00
288	3.01	3.03
320	3.02	3.03

(a)

Series 400, Initial gap 0.5 μm Nominal area 7.672357 sq.mm.		
Pressure	Distortion Coefficient λ (ppm/MPa)	
(MPa)	Sliding	Clamped
32	3.00	*Same as for Sliding Support*
64	2.98	
96	2.96	
128	2.98	
160	2.99	
192	3.02	
224	2.99	
256	3.00	
288	3.00	
320	2.99	

(b)

Table 4.5. Distortion coefficients for the NPL series 400 pressure balance simulated with (a) 1.0 μm and (b) 0.5 μm initial gaps.

the viscosity as well as the contribution of the initial gap being very small and comparable to the elastic distortion at high pressures. A summary of the pressure profiles for various initial gaps is presented in figure 4.11. However, the distortion coefficient was found to be constant at 3.00 ppm/MPa over the operating range (0-320MPa) for both the cylinder sliding and clamped support conditions as shown in Table 4.5.

4.3 Further Studies on the NPL Series 400 Pressure Balance

As this balance is the highest in the range of the NPL pressure balances, further studies were carried out. Namely, the simulation of the balance at higher pressures (up to 1200MPa) and the investigation of the behaviour of the balance with tungsten carbide being the material of construction instead of steel. The cylinder sliding support was chosen for these simulations.

4.3.1 The NPL Series 400 Pressure Balance Simulated up to 1200MPa

As mentioned in chapter 3 (section 3.4), the PVE program was modified to employ a non-iterative procedure for the PV analysis. As a result, the series 400 pressure balance was simulated as being operated at very high pressures (up to 1200MPa). The first step was obviously verifying the results obtained from this version of the PVE program with the one employing the PV iterative method (see discussion in section 3.7). It was found that the distortion coefficients calculated by the two methods (iterative and non-iterative) varied (at the most) by approximately 1% for the series 100, 200, 300 and 400 balances (operating up to 320MPa). This confirmed the validity of the PV non-iterative method. The next step was to actually simulate the series 400 pressure

balance (sliding support) at pressures higher than 320 MPa. This was successfully achieved and up to 1200MPa as described below.

The balance was first simulated at an operating pressure of 800MPa. The minimum operating initial gap was found to be 2.0 μm and the resulting λ 3.17 ppm/MPa. This simulation is presented graphically in figure 4.12 showing the gap and pressure profiles. It required 1000 layers (steps) along the pressure axis compared to the typical 500 layers when simulating the same balance at 320 MPa. The operating pressure was increased simulating the balance at 1000 MPa but at 2.2 μm initial gap as shown in figure 4.13. It can be seen from figures 4.12 and 4.13 that the pressure profile behaved as an "inverted" S-shaped profile which is very different from the previous results (figure 4.11) while the distortion coefficient (λ = 3.25 ppm/MPa) was increased by only 2% compared to the 800MPa result and by 8% compared to the result of 320MPa which can be regarded as remaining fairly constant despite of having a different pressure and gap profiles. The simulation of the series 400 balance at 1000MPa still required 1000 layers. Assuming an operating pressure of 1200 MPa for the series 400 balance, and an initial gap of 2.7 μm, the distortion coefficient was found to be approximately 3.37ppm/MPa, a change of 12% which may be regarded as an extreme case of operation for any present day balance. Note that some 2000 layers were required to simulate the latter case. The pressure and gap profiles were very similar to those in figure 4.13 (the simulation at 1000 MPa).

4.3.2 The NPL Series 400 Pressure Balance Simulated up to 320MPa with Tungsten Carbide

It is useful to see how the series 400 high pressure balance would behave if made of tungsten carbide. For this simulation, a first attempt, reported in reference [25], was

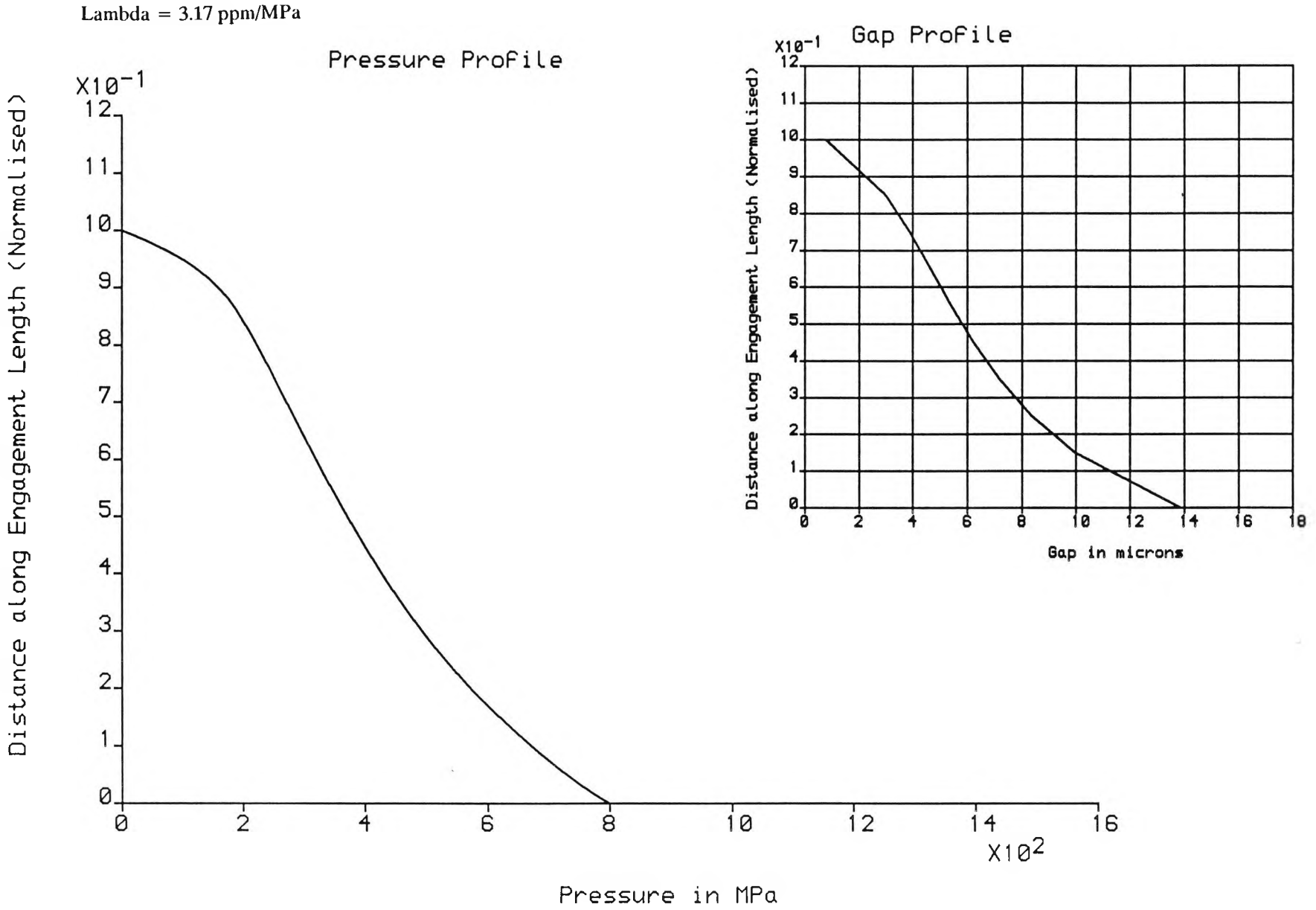


Figure 4.12. Pressure and gap profiles for the NPL series 400 pressure balance simulated at 800 MPa (cylinder - sliding support).

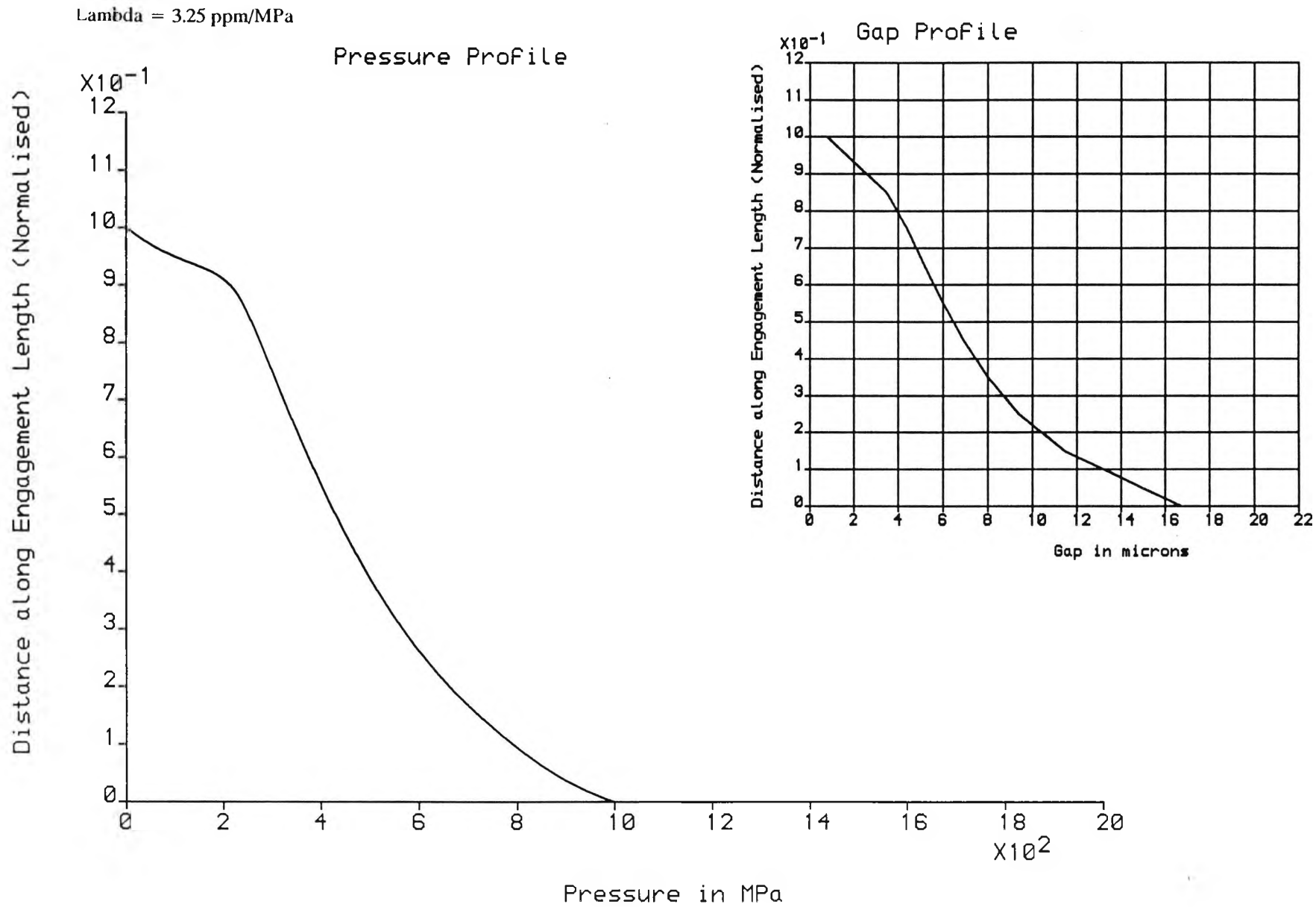


Figure 4.13. Pressure and gap profiles for the NPL series 400 pressure balance simulated at 1000 MPa (cylinder - sliding support).

to assume that the Poisson's ratio of tungsten carbide (being 0.23 compared to 0.3 for steel) will have little contribution to the elastic distortions if compared to the contribution from the difference in the Young's modulus where the latter (i.e tungsten carbide) has a substantially greater Young's modulus of 5.64×10^5 N/mm^2. As the ratio of the two modulii is 0.372, the piston and cylinder distortions (i.e. the 22 runs) already generated using the FEA program, were divided by this ratio and the PVE program was run for the series 400 (now with tungsten carbide as being the material replacing stainless steel). Results for 0.5, 1.0, 2 and 4 μm initial gaps are presented in figures 4.14 to 4.17. Again, these results show a wide variation in gap and pressure profiles but a small variation in the distortion coefficient. The value of λ varies from 1.12 to 1.28 ppm/MPa for the initial gaps quoted above. Another significant change occurs when the viscosity of the oil was "killed" ($z = 0$), where the distortion coefficient was found to be 1.12 ppm/MPa (figure 4.18). This turns out to be approximately the same value as in the low pressure limit ($\lambda =$ 1.15 ppm/MPa). If the result of this analysis is compared with the theory reported in section 5.2.1 of reference [1] (equation 105 and 106 and described as the "normal method" in [19]), the latter will give a λ value of approximately 0.78 ppm/MPa for the series 400 balance made of tungsten carbide. This shows that the (FEA and PVE) analysis results a distortion coefficient value higher by more than 20% compared to the "normal method". This led to the second and more accurate attempt carried out at a later stage (following the publication of reference [25]) as explained below.

The second attempt, and the accurate one, was to reproduce the "unit" load data for the series 400 balance using the FEA program (i.e 22 runs for the piston and cylinder) but assuming the correct value for both the Young's Modulus (5.83×10^5 N/mm^2 provided by NPL) and Poisson's ratio (0.23). These unit data were used by the PVE program to calculate the distortion coefficient. It was found that for an initial gap of

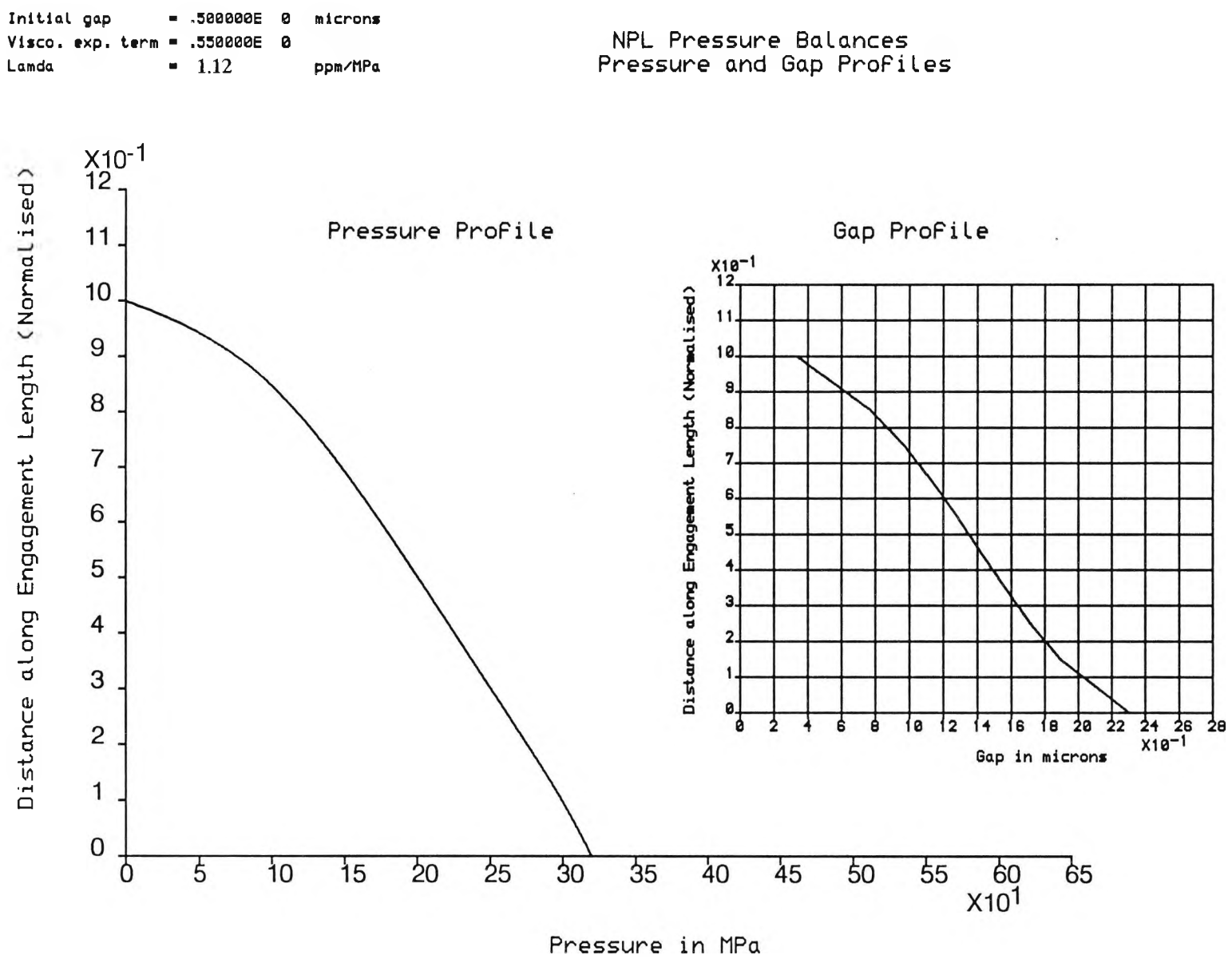


Figure 4.14. Pressure and gap profiles for the NPL series 400 pressure balance simulated as made of tungsten carbide with a 0.5 μm initial gap.

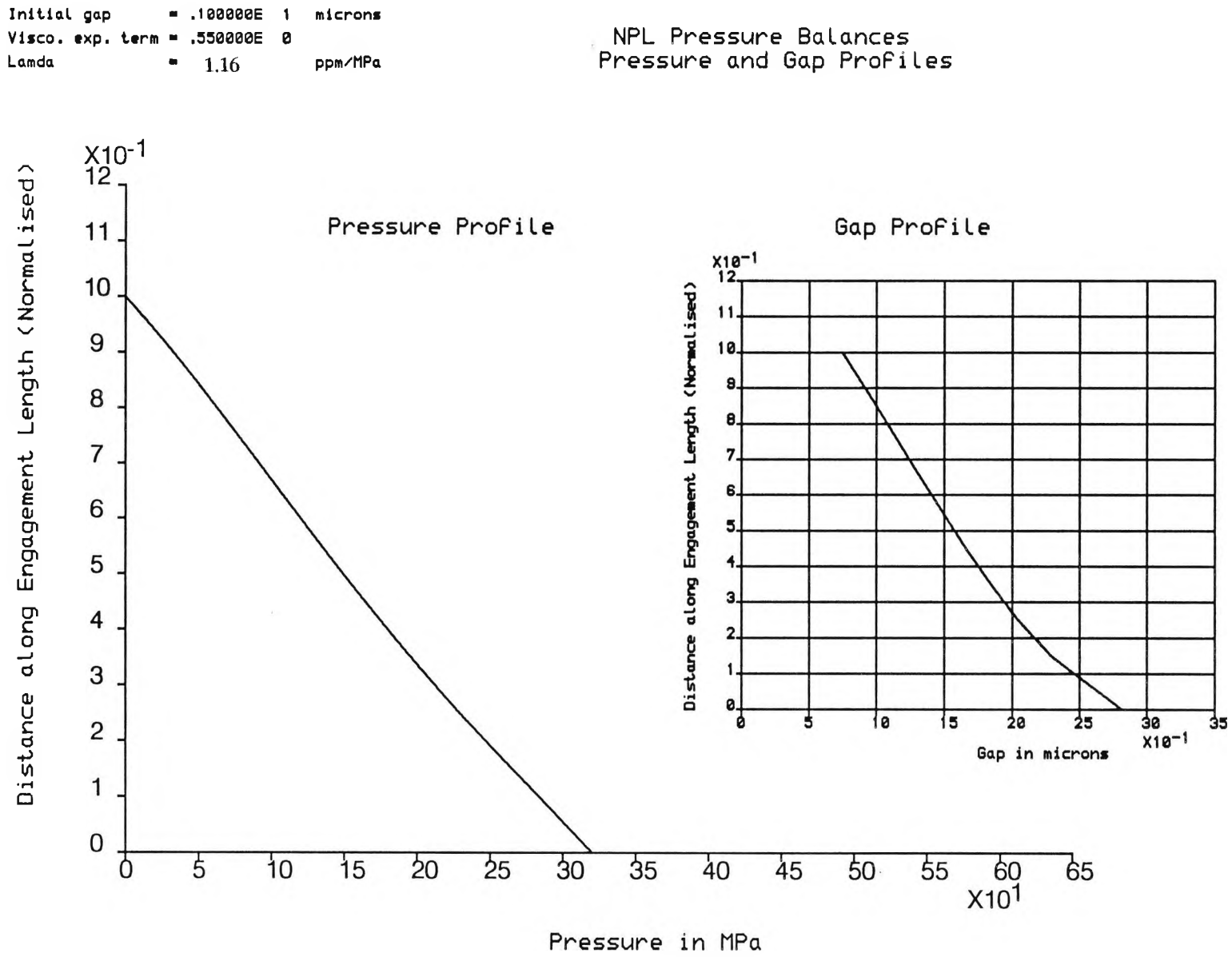


Figure 4.15. Pressure and gap profiles for the NPL series 400 pressure balance simulated as made of tungsten carbide with a 1 μm initial gap.

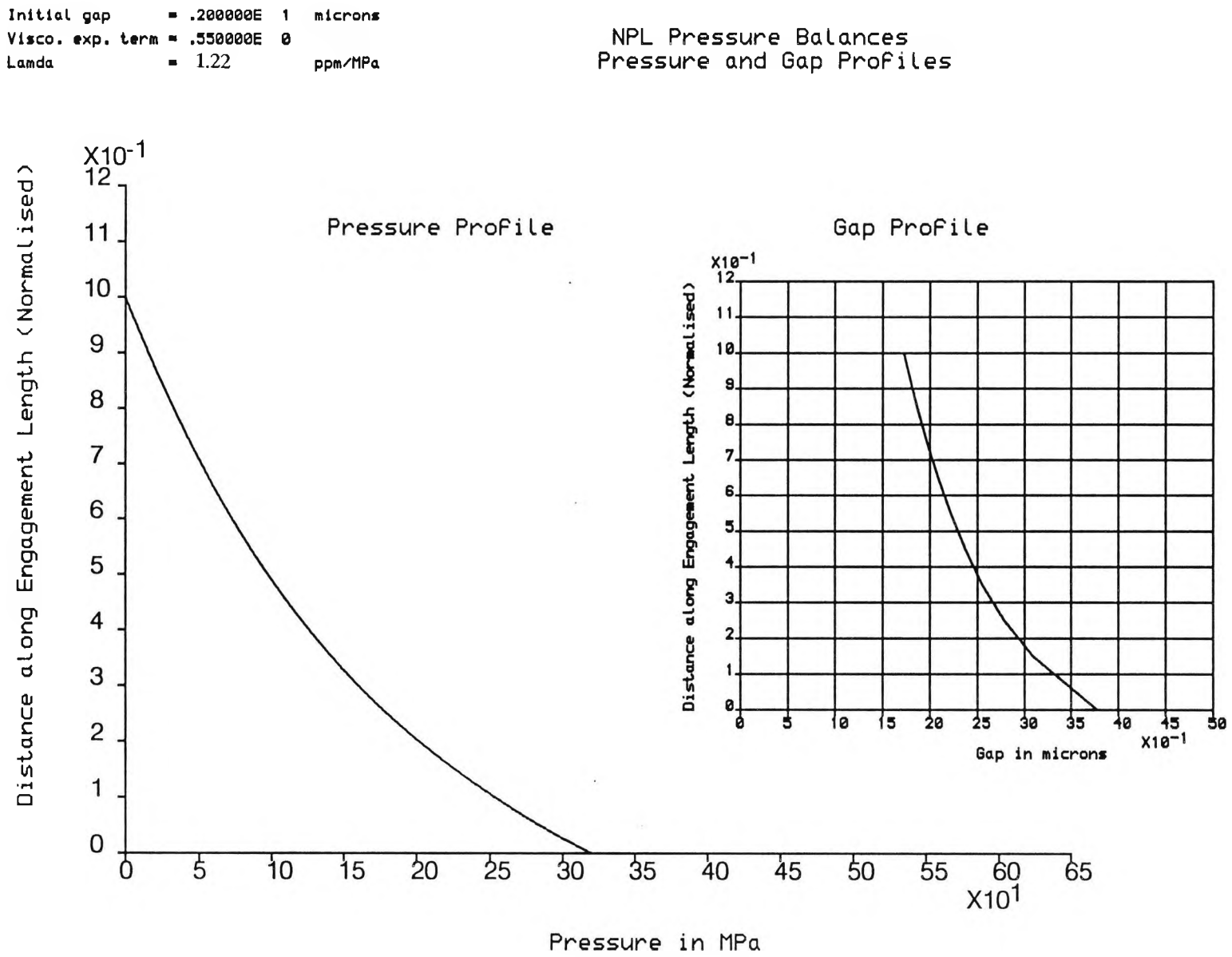


Figure 4.16. Pressure and gap profiles for the NPL series 400 pressure balance simulated as made of tungsten carbide with a 2 μm initial gap.

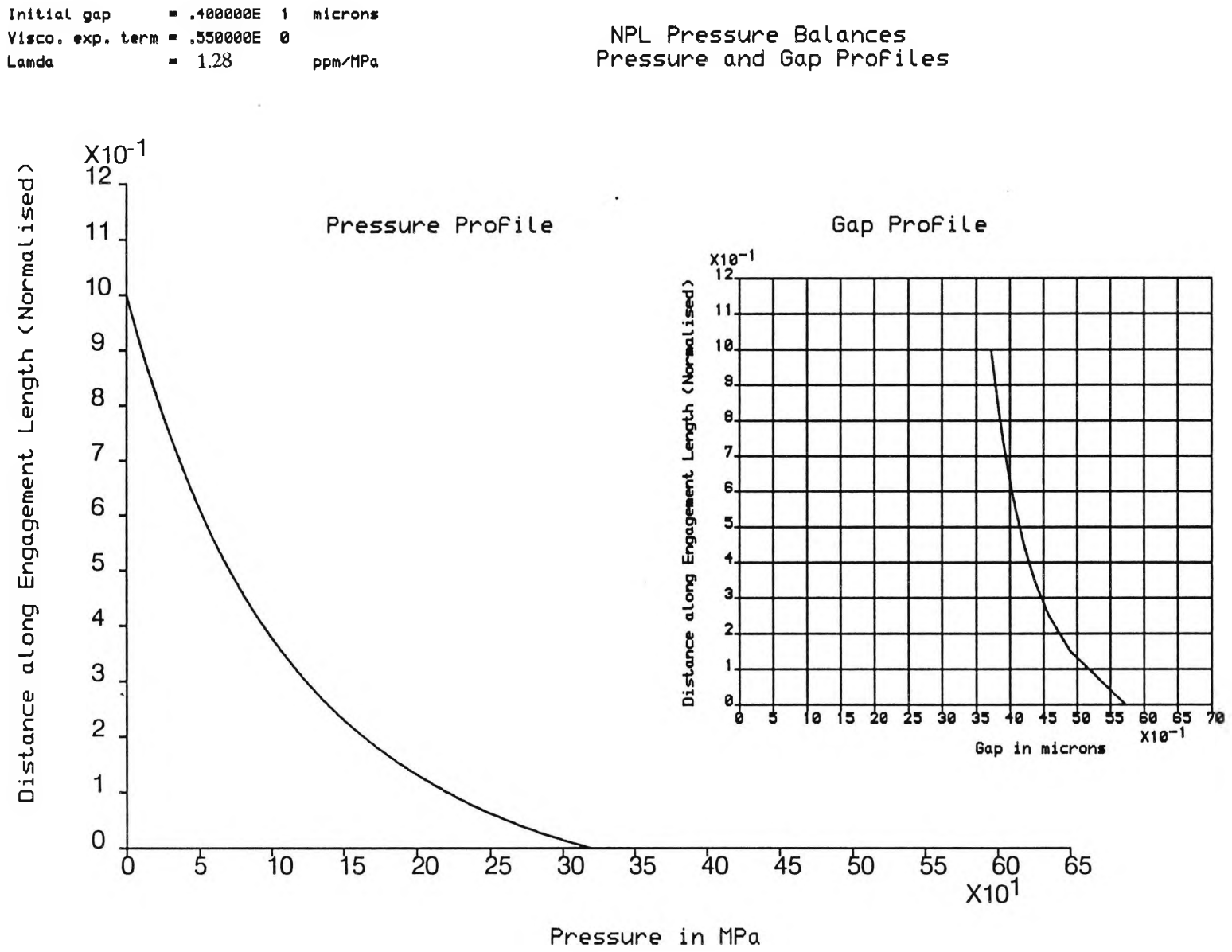


Figure 4.17. Pressure and gap profiles for the NPL series 400 pressure balance simulated as made of tungsten carbide with a 4 μm initial gap.

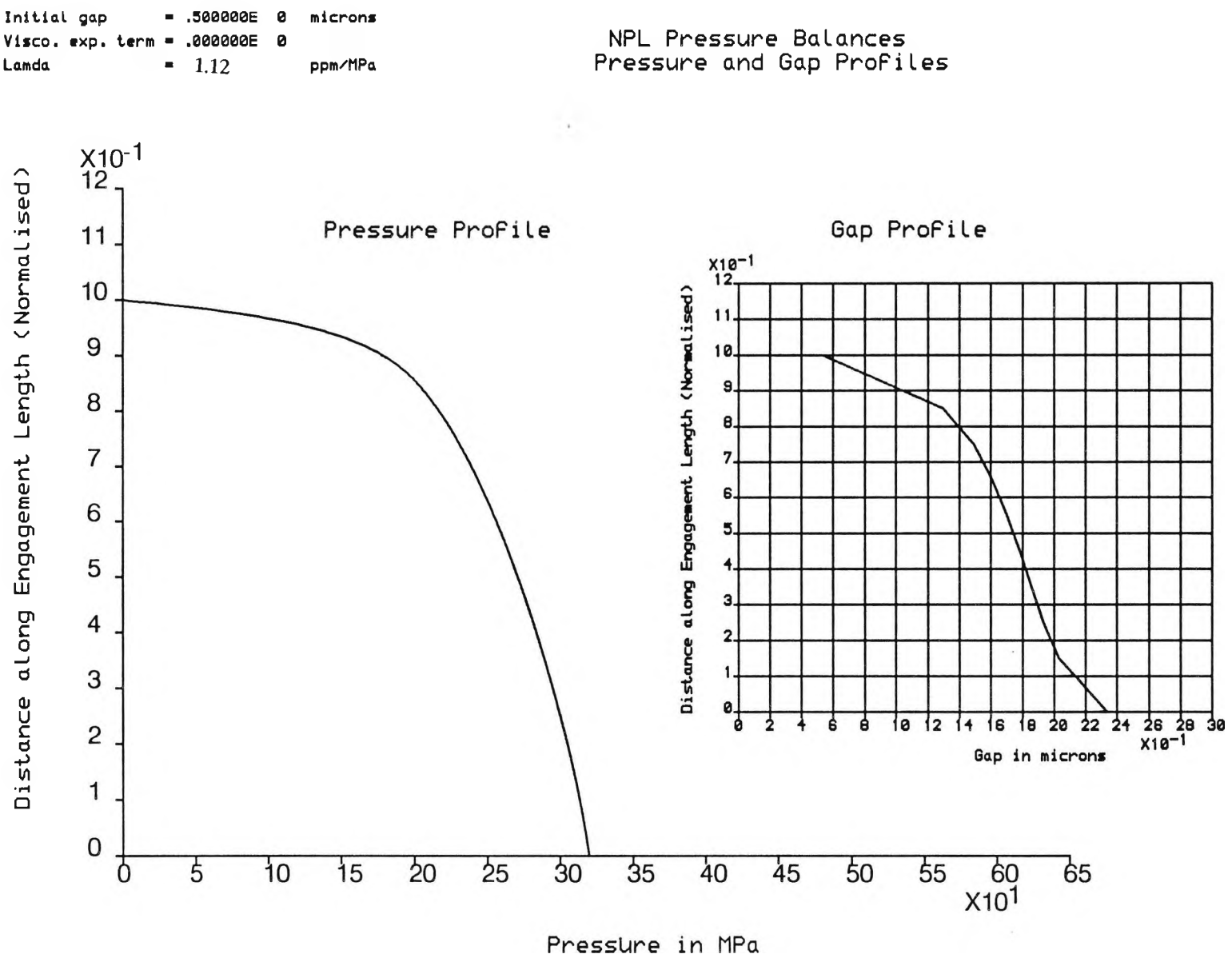


Figure 4.18. Pressure and gap profiles for the NPL series 400 pressure balance made of tungsten carbide (initial gap = 0.5μm) with oil viscosity exponent term z = 0 (therefore viscosity constant).

0.5 or 1.0 μm, the distortion coefficient was approximately 0.8 ppm/MPa. This is still higher than the "normal method" reported in Dadson et al [1], but very much closer compared to the result from the first attempt described above. Further, this result is similar to one obtained from an analysis reported by Molinar et al [2] (for a balance of different geometry) where the latter has calculated a distortion coefficient of 0.747 ppm/MPa.

4.4 The RUSKA Pressure Balances

The methods developed were applied to two of RUSKA pressure balances (2481 and 2470) to show their generality. The RUSKA 2481 is an oil type of balance operating up to 280MPa while the RUSKA 2470 is a gas balance operating up to 17MPa and is a primary standard. These balances are made of tungsten carbide and were simulated as being operated in the gauge mode of operation. Table 4.6 provides their geometrical, material and fluid data (details in Appendix A). Moreover, there was no knowledge of the actual (designed) initial gap in which case it was assumed at a possible value. Note that both RUSKA models are reentrant type of balances.

The schematic diagram for both RUSKA balances is shown in figure 4.19 where the shoulder and base are indicated. As mentioned in section 4.2, added to data needed for the simulation of the balances, the knowledge of the support conditions are required. Similar to the NPL balances, the piston will always possess a sliding support condition.

For the cylinder of the RUSKA 2481 oil balance, two cases of support conditions emerged, defined as:

Geometrical Data	RUSKA 2481	Model 2470
Type of Balance	Oil	Gas
Oil used	Mineral	---
Operating Pressure (MPa)	280.00	17.00
Engagement length (mm)	23.03	22.23
Piston Radius (mm)	1.625	1.625

(a)

Symbol	Description	Unit	Value
E	Youngs modulus	N/mm^2	$5.83x10^5$
ν	Poissons ratio		0.23
η_o	Visc. at amb. press.	mPa.s	25.80
z	Exp. term (eq. 3.13)		0.67

(b)

Table 4.6 Summary of (a) the geometrical and (b) the fluid data for the RUSKA pressure balances

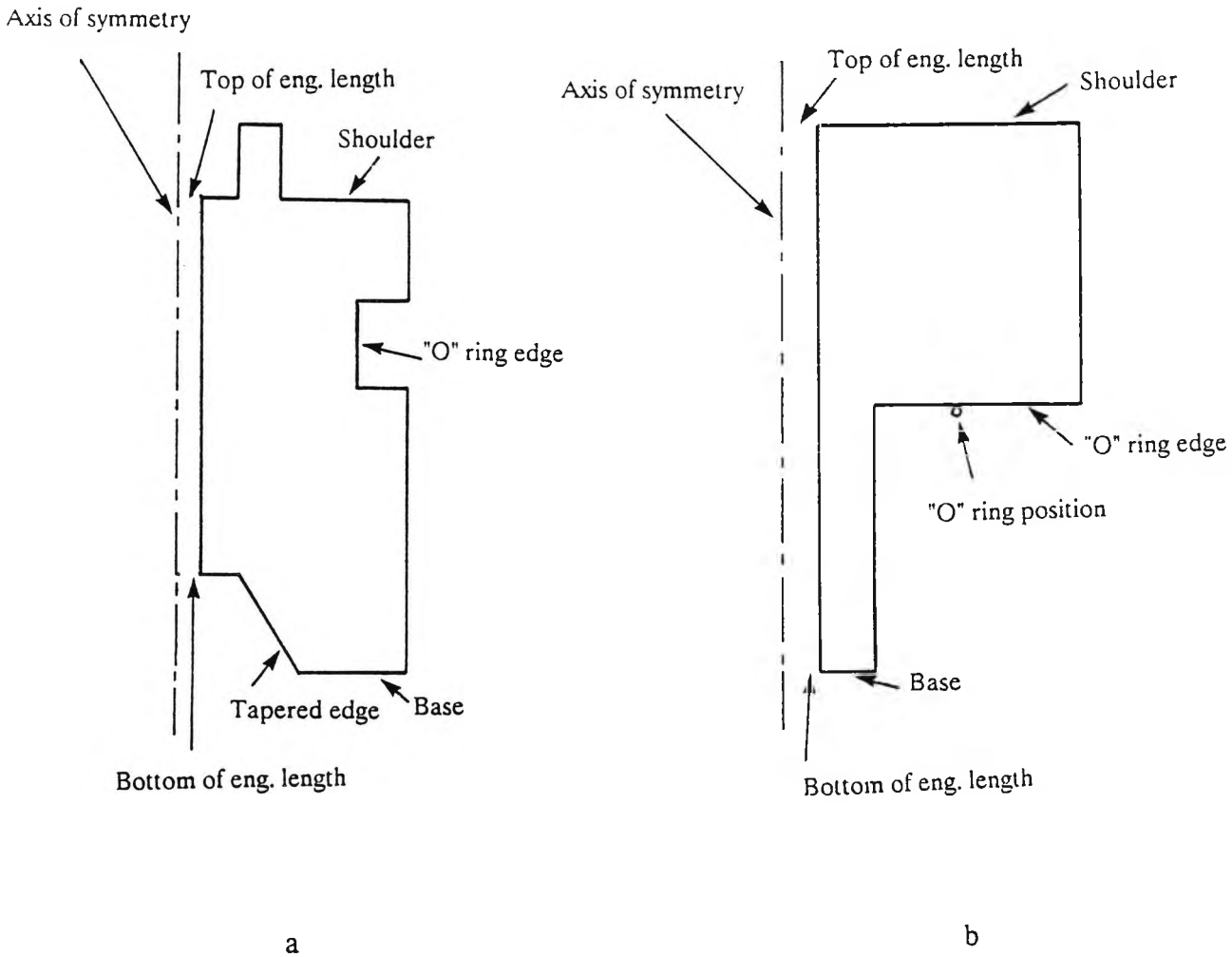


Figure 4.19. Schematic of the cylinders for the RUSKA pressure balances.

(a) RUSKA 2481 (oil operated) and (b) RUSKA 2470 (gas operated).

Case A: Shoulder sliding radially and "O" ring edge clamped,

and

Case B: Shoulder sliding radially and "O" ring edge sliding axially.

With regard to the cylinder of the RUSKA 2470 gas balance, the support conditions were to assume the shoulder sliding radially.

The simulation carried out for these type of balances essentially followed the same procedure as for the NPL balances described in detail for the series 400 (chapter 3, section 3.9 and chapter 4, section 4.2). Regarding the pressure loading for the RUSKA balances, full pressure is applied up to the "O" ring position for the cylinder. The RUSKA oil operated pressure balances were simulated with both the sebacate oil (as used in NPL balances) and a straight mineral oil, the latter being the normal operating oil for the RUSKA 2481 pressure balance.

4.4.1 The RUSKA 2481 Oil Operated Pressure Balance

The radial distortions for 1MPa are shown in figure 4.20 and 4.21 for case A and case B support conditions respectively. The distorted shape is far from linear unlike the NPL pressure balances. This is expected due to the fact that this type of pressure balance is a re-entrant one. Being a high pressure balance, it will experience the variation of pressure due to change in viscosity which in turn will affect the gap and pressure profile along the engagement length which means that an estimation of the elastic distortions at high pressure (from scaling up the 1MPa distortions) will be

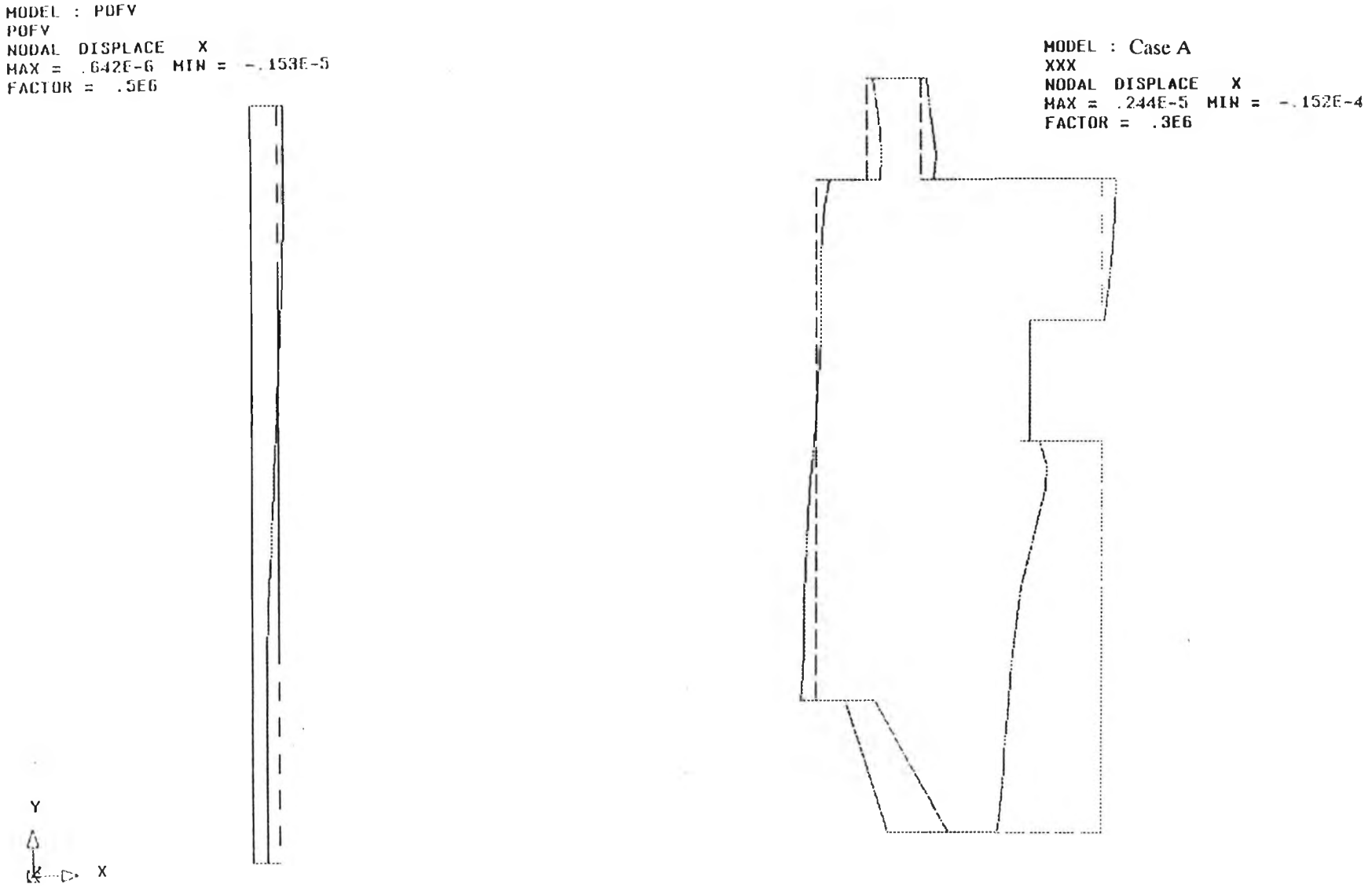


Figure 4.20. Radial distortion at 1 MPa for the RUSKA 2481 (oil operated) pressure balance, cylinder support condition Case A (mag. 5 x 10^5 for piston & 3 x 10^5 for cylinder).

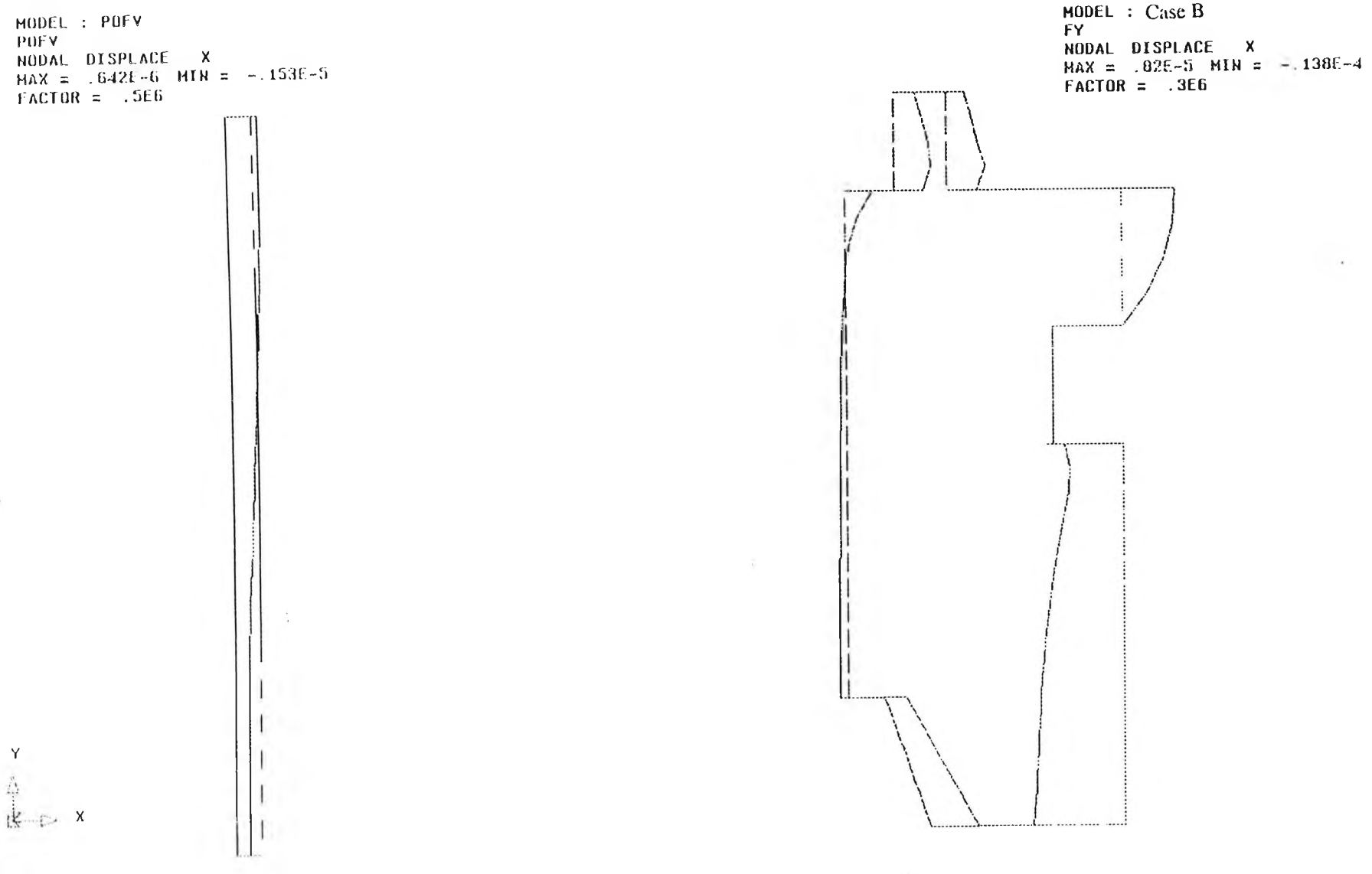


Figure 4.21. Radial distortion at 1 MPa for the RUSKA 2481 (oil operated) pressure balance, cylinder support condition Case B (mag. 5 x 10^5 for piston & 3 x 10^5 for cylinder).

incorrect (for the same reasoning as outlined for the NPL series 300 and 400 pressure balances).

Due to the tapered edge in the design of the cylinder of this balance and being reentrant, the balance will experience upward forces (in the axial direction). This is another significant difference between the RUSKA and NPL balances.

Using the PVE program, support conditions **case A** with 2.0 μm initial gap and **case B** with initial gaps of 1.5 and 2.0 μm were simulated using the mineral oil.

The pressure and gap profiles for both support conditions A and B are shown in figure 4.22. Both profiles are different from the simple design of the series 400 NPL pressure balance. For the RUSKA design, case A, the pressure drops rapidly over the bottom half of the engagement length. This is in contrast to the NPL design where the rapid drop was over the top half of the engagement length. The gap profile, however, does not follow the pressure profile due to fact that this design is a reentrant one where there will be a full operating pressure acting on the outer edge of the cylinder up to the "O" ring position. The gap profile is unusual since it has a turning point (minimum) at one third way up the engagement length.

For support condition case B two initial gaps of 1.5 μm and 2.0 μm were simulated and the pressure and gap profiles were found to be very similar. Results for 2 μm initial gap are shown in figure 4.22. The pressure profile is similar to that of case A but the drop is less severe whereas the gap profile for this case is much smoother compared to that of case A.

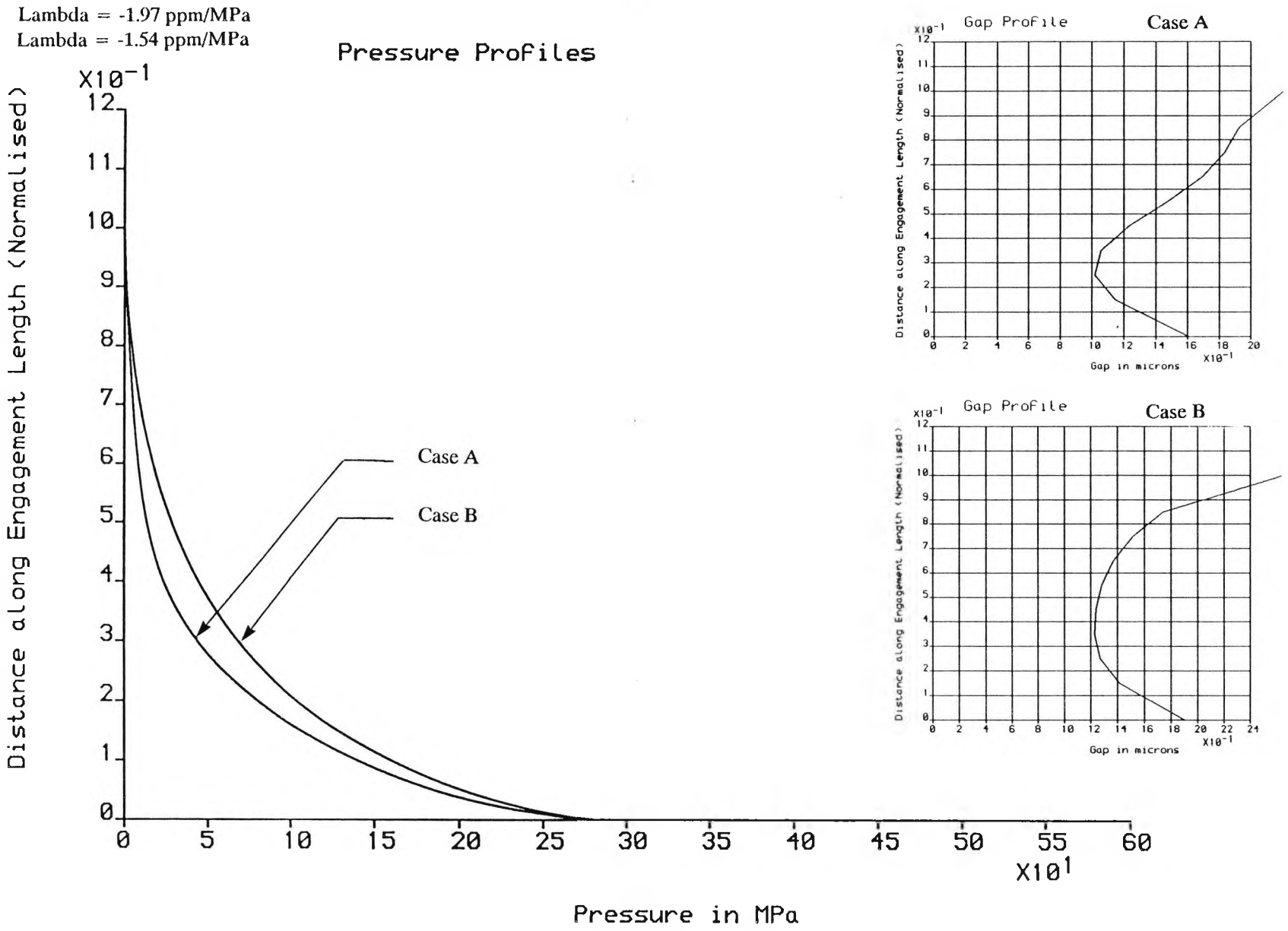


Figure 4.22. Pressure and gap profiles for the RUSKA 2481 (oil operated) pressure balance (cylinder support conditions Case A and Case B).

For case A, the distortion coefficient was found to be varying from -0.70 ppm/MPa at 28MPa to -1.96 ppm/MPa at 280MPa. The distribution of the distortion coefficient over the operating range of the balance versus the operating pressure is shown in figure 4.23 and it can be seen that it is neither constant nor even linear. As regard to case B, the distortion coefficients for both initial gaps (1.5 and 2.0 μm) were obtained and found to be the same to within 1% percent. For an initial gap of 1.5 μm, the distortion coefficient varied between -0.74 ppm/MPa and -1.49 ppm/MPa while for the 2.0 μm initial gap λ varied between -0.73 ppm/MPa and -1.54 ppm/MPa over the operating pressure range (28MPa to 280MPa). As shown in figure 4.23, the variation of the distortion coefficient is of a similar behaviour to that of the case A. Results for 2 μm initial gap are provided in Table 4.7a for cases A and B.

Simulating the RUSKA 2481 for both cases A and B but with the sebacate oil (similar to the one used in NPL balances), the pressure and gap profiles were found to be very similar as in figure 4.22. The λ profiles are shown in figure 4.23 and it can be seen that they are less steep for both support conditions A and B when comparing with the mineral oil. Also, the lower limit of λ (at 28 MPa) remained unchanged regardless of the support condition or the type of the oil used.

The λ distributions in figure 4.23 need to be fitted with a two-degree polynomial. Consequently, it was found that the effective area, A, must have a cubic term, so:

$$A = A_o (1 + aP + bP^2 + cP^3)$$

a, b and c are constant coefficients.

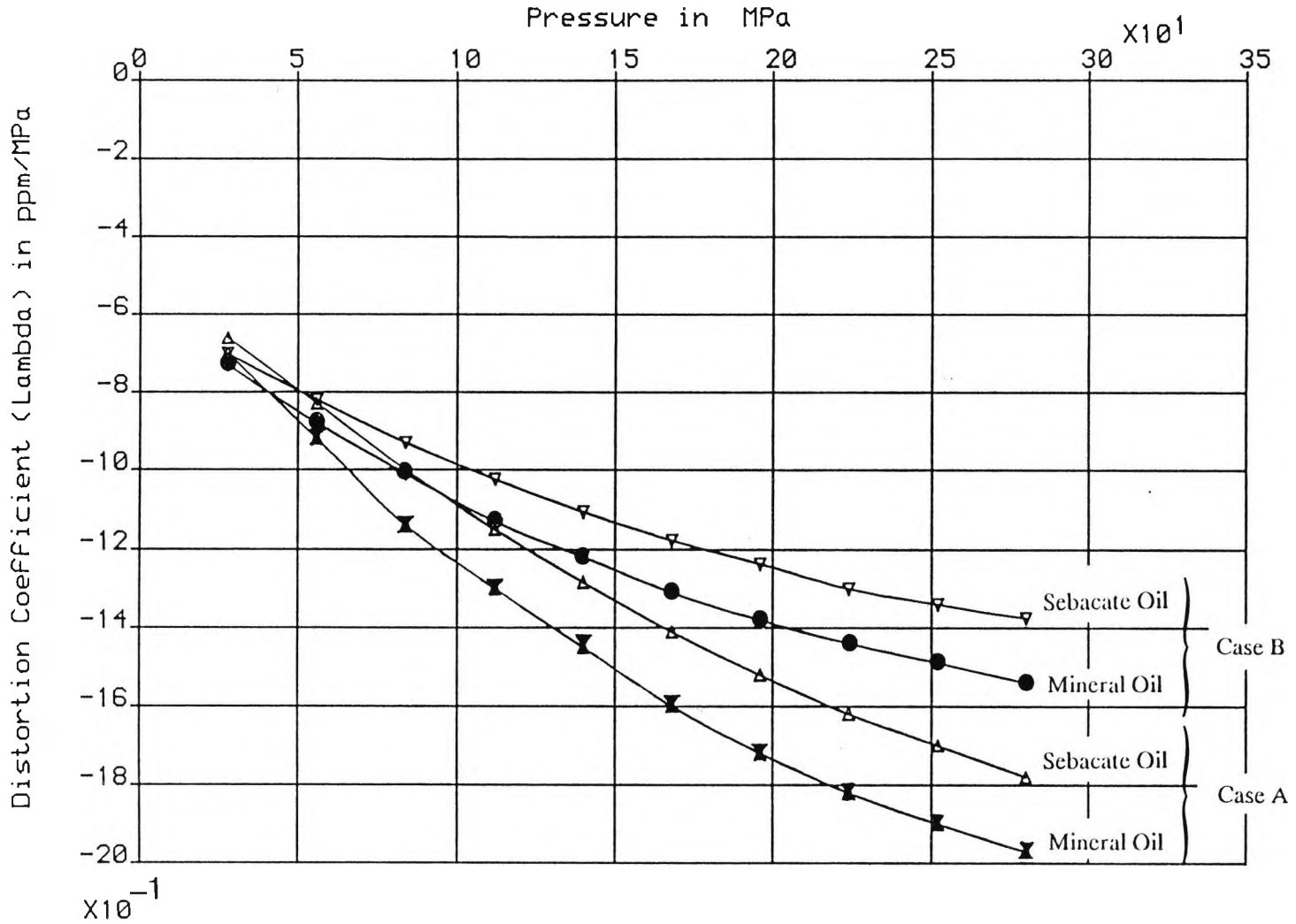


Figure 4.23. Variation of the distortion coefficient (λ) for the RUSKA 2481(oil operated) pressure balance.

RUSKA 2481 (Oil), Initial gap 2.0 μm		
Pressure (MPa)	Distortion Coefficient λ (ppm/MPa)	
	Case A	Case B
28.0	-0.70	-0.73
56.0	-0.92	-0.88
84.0	-1.14	-1.01
112.0	-1.30	-1.13
140.0	-1.45	-1.22
168.0	-1.60	-1.31
196.0	-1.72	-1.38
224.0	-1.82	-1.44
252.0	-1.90	-1.49
280.0	-1.97	-1.54
Nominal Area 8.305976549 sq. mm.		

(a)

Constants@ of the RUSKA 2481 λ Profile Fit				
Constant	Case A		Case B	
	Mineral Oil	*Sebacate Oil*	*Mineral Oil*	*Sebacate Oil*
a	-0.47	-0.47	-0.58	-0.58
b	-8.81×10^{-3}	-6.98×10^{-3}	-5.74×10^{-3}	-4.70×10^{-3}
c	1.24×10^{-6}	8.24×10^{-6}	8.41×10^{-6}	6.60×10^{-6}

$@_{\lambda} = a + bP + cP^2$

{where λ in ppm/MPa, P in MPa, **a** in ppm/MPa, **b** in ppm/(MPa)^2 and **c** in ppm/(MPa)^3 }.

(b)

Table 4.7. The RUSKA 2481 oil operated pressure balance.(a) Distortion coefficients. (b) Coefficients for the λ distribution fit .

The fitted coefficients for the λ curves are given in Table 4.7b for cases A and B using both the mineral and the sebacate oils. Using the given values of a, b and c, the curves were fitted to better than 2%.

Finally, the RUSKA 2481 was simulated for the conditions as in case B (with the mineral oil) but the "O" ring seal was assumed to be at the bottom of the cylinder engagement length. This investigation was to simulate a non-reentrant design similar to the NPL series 400 pressure balance. For this case (a "non- reentrant" RUSKA 2481), the distortion coefficient was found to be constant at a value of 0.76 ppm/MPa over the operating pressure range.

It can be concluded that for cases A and B, the distortion coefficient experienced a variation due to the existence of upward forces occurring at the bottom of the cylinder (due the tapered design) while the simulation of the same balance as a non-reentrant one showed a constant λ (with a positive value). This is in good agreement with the theory reported in Dadson et al [18] since for the non-reentrant (simple) design there will be no forces (i.e. no stresses in the y , axial, direction) along the cylinder.

4.4.2 The RUSKA 2470 Gas Operated Pressure Balance

The radial distortions for this balance (gas) operated at 1MPa are shown in figure 4.24. The distorted shape is somewhat similar to the RUSKA 2481 oil balance. For an initial gap of 1.0 μm (actual implemented gap unknown), the pressure and gap profiles are presented in figure 4.25 using the PVE program. The pressure drop along the engagement length is very similar to the simple design of the series 200 NPL pressure balance being not very far from linear. The gap profile, however, does not follow the

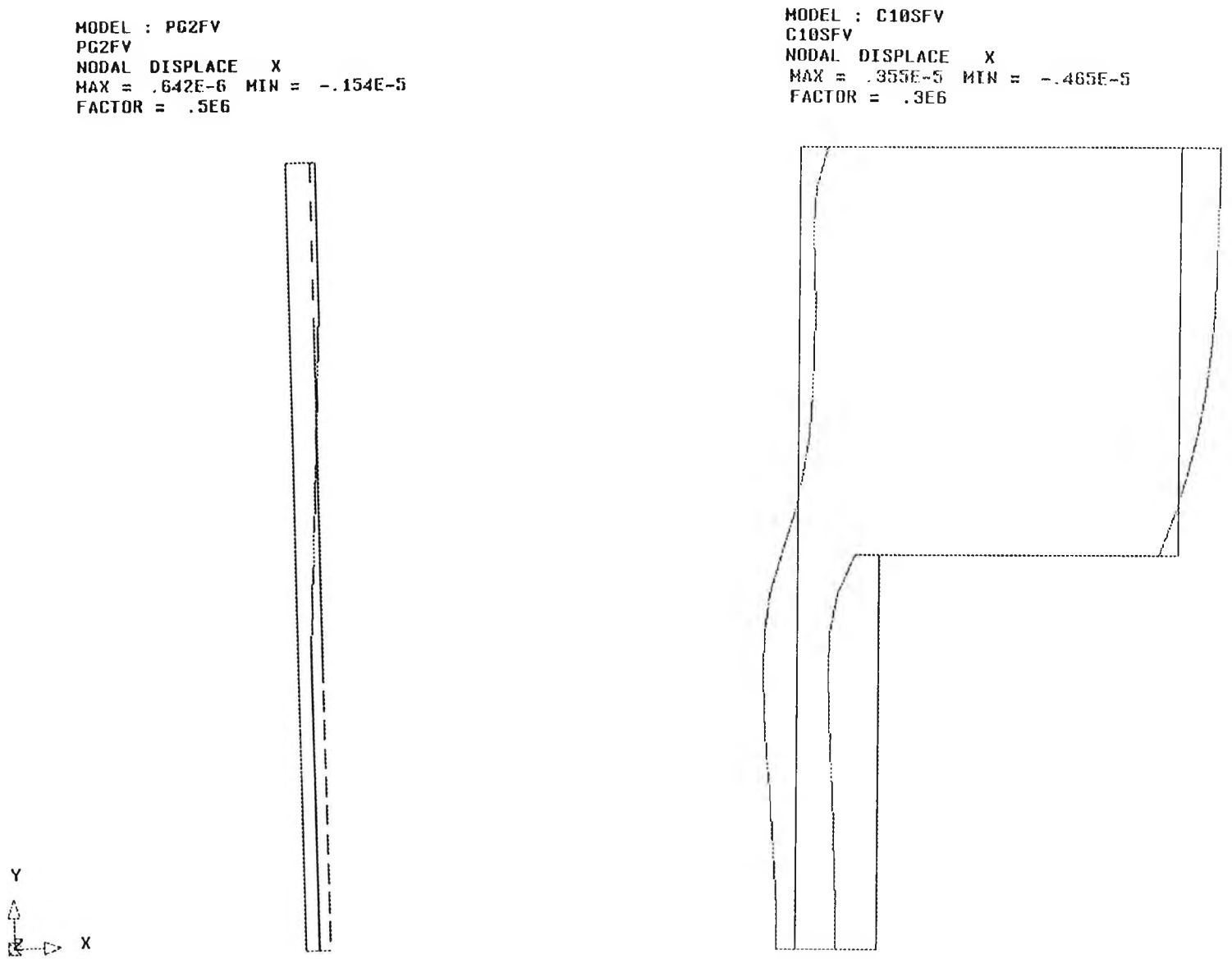


Figure 4.24. Radial distortion at 1 MPa for the RUSKA 2470 (gas operated) pressure balance (magnification 5 x 10^5 for piston & 3 x 10^5 for cylinder).

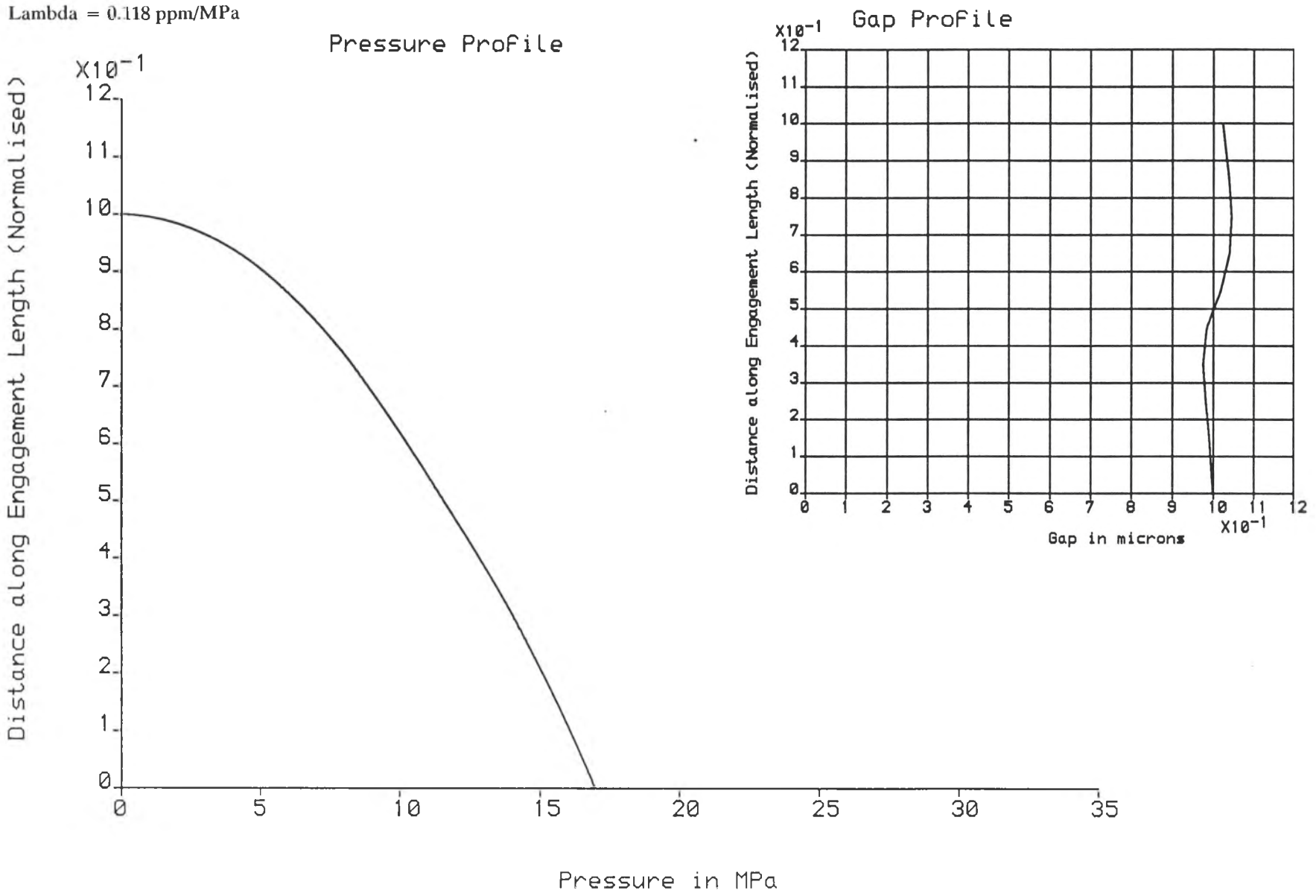


Figure 4.25. Pressure and gap profiles for the RUSKA 2470 (gas operated) pressure balance.

pressure profile but it is fairly **constant** although this balance is a reentrant one. The distortion coefficient was found to be **constant** at a value of 0.14 ppm/MPa over the operating range of the balance (1.7 MPa to 17 MPa). These results are presented in Table 4.8.

A sensitivity test was carried out to investigate the effect of the "O" ring position on the distortion coefficient. The first attempt was to simulate the balance having the "O" ring at the most left and most right of the "O" ring edge (see schematic figure 4.19). This required the reproduction of the 22 run for the "unit" load data using the FEA program. The application of the 1 MPa radial distortions for these two cases were very similar to the "O" ring in its normal position. Likewise the pressure and gap profiles obtained from the PVE program showed relatively no change when comparing these profiles when having the "O" ring at the left, middle (normal) or right positions.

However, an interesting result was obtained as regard to the distortion coefficients. Firstly, the distortion coefficient remained almost constant with operating pressure for the left or right position, but it showed a change of sign, a **negative average** value of -0.055 ppm/MPa when having the "O" ring at the left position to a **positive average** value of 0.18 ppm/MPa in the case of having the "O" ring at the right position. This leads to the conclusion that this (RUSKA 2470) can deliver a virtually **zero** distortion coefficient by adjusting the "O" ring position along its edge.

A further investigation included the simulation of the RUSKA 2470 gas balance but assuming it operates with oil (of the same type as in the NPL series of balances). This simulation showed a distortion coefficient of about -1.0 ppm/MPa which is significantly different from that when operating with gas.

RUSKA 2470 (Gas), Initial Gap 1.0 μm	
Pressure (MPa)	Distortion Coefficient λ (ppm/MPa)
1.70	0.12
3.40	0.16
5.10	0.16
6.80	0.16
8.50	0.16
10.20	0.15
11.90	0.14
13.60	0.14
15.30	0.13
17.00	0.12
Nominal Area 8.300871462 sq. mm.	

Table 4.8. Distortion coefficients for the RUSKA 2470 gas operated pressure balance.

Finally, as for the RUSKA 2481 oil type of balance, the RUSKA 2470 (gas balance) was simulated assuming a non-reentrant design. That is assuming that the "O" ring seal is at the bottom of the cylinder engagement length. For the same 1.0 m initial gap, the distortion coefficient was constant but now at a value of +1.00 ppm/MPa. Further, simulating this case as if the RUSKA 2470 is filled with oil (as used in the NPL balances), λ was found to be 1.05 ppm/MPa, again unchanged.

4.5 Summary of the Results and Conclusions

Having described the detailed results for all the simulated pressure balances, the summary of the distortion coefficients are re-presented (for convenience) in Table 4.9 for the NPL pressure balances and Table 4.10 for the RUSKA pressure balances.

4.5.1 The NPL Pressure Balances

For the NPL pressure balances, the sliding support is assumed to be the normal operating condition. The distortion coefficient showed a λ constant (to within a few percent) at 3.30 ppm/MPa for the series 100 and 200 balances, and 3.00 ppm/MPa for the series 300 and 400 balances. However, a clamped support condition may arise due to variations in the assembly. That is if the torque applied to the cylinder retaining collar is too large, a frictional contact could prevent the cylinder base and shoulder from sliding freely. For this (clamped) condition, there is a significant decrease of 30% in λ for series 100 and a decrease of 12% for the series 200 balances. For the series 300 and 400 balances, there is no significant change in λ between the clamped and sliding support conditions. However, there is a noticeable change in λ when simulating the series 400 high pressure balance at an extended range of its operation (up to 1200 MPa)

where λ was found to be approximately 3.37 ppm/MPa. Further, the simulation of the series 200 balance but made of tungsten carbide resulted in a distortion coefficient of 0.94 ppm/MPa at 20 MPa and remained unchanged for lower operating pressures, compared to 1.5 ppm/MPa, an experimental value of λ for the same balance (provided by NPL [26]). Finally, the latter simulation was carried out on the NPL series 400 balance (i.e. made of tungsten carbide) and the distortion coefficient was found to be constant at 0.78 ppm/MPa over the operating pressure range of the balance.

In this study, the detailed simulation of the series 400 high pressure balance, in particular, has shown the great value of the PVE program developed and the methodology on which it is based. With graphical display of the pressure and gap profiles, considerable insight into the operation of a balance can be obtained. For example, figure 4.11 showed a wide variation in these profiles but with a small variation in λ (constant to within 7%). This result validates the assumption of a constant distortion coefficient in the operation of this (high pressure) balance. In fact, it was one aim of the work presented in this thesis to verify (or otherwise) the validity of the normally used effective area equation for all the NPL pressure balances and in particular the high pressure one.

The simulation of any balance (once characterised with "unit" load data using the FEA program) takes only a few minutes using the PVE program running on a Sun/SPARC workstation. Repeated runs allow one to build up tables for the distortion coefficient versus the operating pressure range and even beyond the normal operating conditions of a balance.

It is of interest to note that the actual form of the pressure profile along the engagement length of a simple geometry balance (such as the NPL pressure balances) can vary considerably. Simulated results indicated that the actual profile will be a compromise between the non uniformity of the gap profile and the severity of oil viscosity changes with pressure. This can be summarised as presented in Table 4.11.

Dadson et al [1] provided an analytic formula for calculating the distortion coefficient for pressure balances of simple geometry. For a balance with the piston and cylinder made of the same material, λ is given by [1]:

$$\lambda = \frac{1}{E}\left\{ 2\nu + \frac{R^2}{R'^2 - R^2} \right\} \qquad (4.1)$$

where E is the Young's modulus, ν Poisson's ratio, R cylinder inner radius and R' the cylinder outer radius. Applying this formula to the NPL balances, results agree well with those obtained from the simulation in this thesis for the series 100 to 400 cylinder sliding support conditions (Table 4.12).

The experimental λ values for all NPL pressure balances is (4.0±0.4) ppm/MPa. This value (provided by NPL [26]) is obtained by the similarity method applied to the series 300 balance and by cross-floating for the lower and higher balances. This is significantly different from the values in Table 4.12 and points to the need for further study.

4.5.2 The RUSKA Pressure Balances

The RUSKA 2481 oil and RUSKA 2470 gas operated balances are of reentrant design with a fairly complex geometry. In general, with complex geometries, the evaluation

of the distortion coefficient cannot be expressed by simple analytic expressions. So, here the value of a generally applicable method such as developed in this thesis, becomes particularly significant.

The results for the distortion coefficients (summarised in Table 4.10) show that the RUSKA 2481 balances has a non-linear effective area characteristic. The fitted characteristic for $\lambda = a + bP + cP^2$ is good to better than 2% (with the given values of a, b and c).

Taking case B as the more likely balance boundary conditions, one can compare results with those provided by the manufacturer [27] who give a=-1.86 ppm/MPa, b=-3.5x10^{-3} ppm/(MPa)^2 and with *no* c term. There is therefore a difference both in shape and the mean value of λ. There is approximately a factor of two difference for mean λ between the results presented in this thesis and those quoted by RUSKA. Clearly, this needs further exploration. Also, simulating a non-reentrant version of this balance gave a **constant** λ (of 0.76 ppm/MPa) indicating the possibility that the reentrant design has overcompensated on λ to produce a rather large negative and variable value.

With regard to the RUSKA 2470 gas operated pressure balance, the distortion coefficient was found to be essentially constant over the operating pressure range at a small average value of 0.14 ppm/MPa (details presented in Table 4.10). Further simulation showed that the distortion coefficient could be made exactly zero by careful positioning of the "O" ring seal. The manufacturer claims that this design gives a zero distortion coefficient [27]. The results in this thesis indicate this to be a possibility.

Series 100, Initial gap 1.5 μm, Nominal area 490.9327 sq. mm.		
Pressure	Distortion Coefficient λ in ppm/MPa	
MPa	Sliding	Clamped
0.50	3.25	2.28
1.00	3.26	2.28
1.50	3.26	2.29
2.00	3.27	2.29
2.50	3.27	2.30
3.00	3.27	2.31
3.50	3.28	2.31
4.00	3.28	2.32
4.50	3.29	2.32
5.00	3.29	2.33

(a)

Series 200, Initial gap 1.0 μm, Nominal area 122.7381 sq. mm.		
Pressure	Distortion Coefficient λ in ppm/MPa	
MPa	Sliding	Clamped
2.00	3.27	2.83
4.00	3.27	2.84
6.00	3.27	2.89
8.00	3.28	2.87
10.0	3.28	2.89
12.0	3.29	2.91
14.0	3.30	2.92
16.0	3.31	2.94
18.0	3.31	2.96
20.0	3.32	2.97

(b)

Series 300, Initial gap 0.5 μm, Nominal area 30.68452 sq. mm.		
Pressure	Distortion Coefficient λ in ppm/MPa	
MPa	Sliding	Clamped
8.00	3.04	2.96
16.0	3.03	2.97
24.0	3.04	2.98
32.0	3.05	3.01
40.0	3.06	3.02
48.0	3.07	3.04
56.0	3.08	3.05
64.0	3.09	3.06
72.0	3.09	3.07
80.0	3.10	3.07

(c)

Table 4.9. Distortion coefficients for NPL pressure balances for cylinder sliding and clamped support. (a) Series 100, (b) Series 200, (c) Series 300, (d) Series 400 at 1.0 μm *initial gap, (e) Series 400 at* 0.5 μm *initial gap.*

Series 400, Initial gap 1.0 μm, Nominal area 7.674814 sq. mm.		
Pressure	Distortion Coefficient λ in ppm/MPa	
MPa	Sliding	Clamped
32.0	3.03	3.00
64.0	3.01	2.99
96.0	3.00	2.99
128	3.00	2.99
160	3.00	2.99
192	3.00	2.99
224	3.01	3.00
256	3.01	3.00
288	3.01	3.03
320	3.02	3.03

(d)

Series 400, Initial gap 0.5 μm, Nominal area 7.672357 sq. mm.		
Pressure	Distortion Coefficient λ in ppm/MPa	
MPa	Sliding	Clamped
32.0	3.00	*same as for sliding support*
64.0	2.98	
96.0	2.96	
128	2.98	
160	2.99	
192	3.02	
224	2.99	
256	3.00	
288	3.00	
320	2.99	

(e)

RUSKA 2481 (Oil), Initial gap 2.0 μm		
Pressure	Distortion Coefficient λ (ppm/MPa)	
(MPa)	**Case A**	**Case B**
28.0	-0.70	-0.73
56.0	-0.92	-0.88
84.0	-1.14	-1.01
112.0	-1.30	-1.13
140.0	-1.45	-1.22
168.0	-1.60	-1.31
196.0	-1.72	-1.38
224.0	-1.82	-1.44
252.0	-1.90	-1.49
280.0	-1.97	-1.54
Nominal Area 8.305976549 sq. mm.		

(a)

RUSKA 2470 (Gas), Initial Gap 1.0 μm	
Pressure (MPa)	Distortion Coefficient λ (ppm/MPa)
1.70	0.12
3.40	0.16
5.10	0.16
6.80	0.16
8.50	0.16
10.20	0.15
11.90	0.14
13.60	0.14
15.30	0.13
17.00	0.12
Nominal Area 8.300871462 sq. mm.	

(b)

Table 4.10. Distortion Coefficients for the RUSKA pressure balances. (a) RUSKA 2481 (oil) and (b) RUSKA 2470 (gas).

max / min viscosity ratio	max / min gap ratio	
	small ($\cong$ 1)	large (> > 1)
small ($\cong$ 1)	*linear*	*top end biased*
large (> > 1)	*bottom end biased*	*top, bottom, linear or combination*

Table 4.11 Qualitative pressure profile distributions in pressure balances

NPL Balances Series	Distortion Coefficient (ppm/MPa)	
	Formula*	This Work@
100	3.26	3.27
200	3.07	3.29
300	2.91	3.07
400	2.87	3.00

* R′ in eq. 4.1 taken as outermost radius of NPL cylinders.

@ Average values taken from table 4.9.

Table 4.12 Comparison of analytical and numerical values for the NPL pressure balances.

CHAPTER 5

Conclusions

The work presented in this thesis involved the development and application of mathematical models to pressure metrology; both pressure sensing and pressure generating instruments.In the early stages of the work, the use of analytic and numerical (Finite Element, FE) methods were applied to the modelling of pressure sensors in a relatively new area, namely, the optically driven resonant structures. Also reviewed was the use of the mathematical models applied to industrial design of pressure sensors (using conventional techniques based on elastic elements). Regarding the optically driven sensors, a device based on quartz-crystal resonating structure was successfully modelled to calculate various parameters such as the maximum deflection, resonant frequency and the frequency/load relationship. Results showed good agreement with the experimentally investigated prototype device in contrast to theoretical values previously determined by other workers (details can be found in chapter 2, section 2.4 and a publication appended to this thesis). This work naturally led to tackling the modelling of an instrument used in pressure standards metrology: the *pressure balance*.

For modelling the pressure balance, a novel approach was implemented by characterising a given pressure balance with some "base" data which incorporated the elastic behaviour of the balance and this data was subsequently used to simulate the balance behaviour incorporating the pressure-viscosity effects which become apparent

particularly at high pressures (above 100 MPa). A purpose built program was developed to simulate and present graphically the behaviour of **any** pressure balance. The modelling was essentially based on the use of the Finite Element (FE) method and other numerical techniques.

The model was applied to pressure balances of simple geometries (such as the NPL primary pressure balances, series 100 to series 400, covering an operating pressure range up to 4 to 320 MPa respectively) and more complex geometries including the reentrant gas and oil operated balances (such as the RUSKA balances).

Results were in good agreement with well established theoretical work based on analytical methods and reported by Dadson et al [1]. The latter work showed that a pressure balance can be characterised by relating its effective area to the applied pressure by a linear term in pressure with λ, being the distortion coefficient. Although Dadson showed, on theoretical grounds, that the distortion coefficient is for simple type of balances **constant**, independent of the piston-cylinder gap profile or rheological properties, yet it (i.e. the work) noted that this may **not** be the case for certain conditions at very high pressures.

In this thesis, by application of the models to the NPL series of balances, it was shown that the distortion coefficient of a pressure balance of a simple geometry, remains essentially **constant** at about 3.00 ppm/MPa (for details, see section 4.5.1 of chapter 4). However, at very high pressures applied to these (simple type) balances, the distortion coefficient deviates noticeably to a value of 3.25 ppm/MPa at 1000 MPa and 3.37ppm/MPa at 1200 MPa applied pressures.

For the series 300 (operating up to 80 MPa) and the series 400 (320 MPa) NPL high pressure balances, the distortion coefficient, λ, was constant (at 3.00 ppm/MPa) despite significant variations in the pressure and gap distributions along the engagement length of the balance. In addition, for the series 400 balance, results show a significant contribution to the pressure profile due to the oil pressure-viscosity variations along the engagement length.

The experimental values (of 4 ppm/MPa for all NPL balances) were obtained by tracing back to cross-floating experiments with the series 300 balance using the similarity method. Since the modelling developed in this thesis can be applied to simulate a pressure balance design made of different materials (e.g. steel or tungsten carbide) it is possible to test the validity of the similarity method. This needs further exploration.

On the other hand, the application of the methodology developed (in the models) to other types of balances of more complex geometries led to some interesting results. These were shown by applying the models to the RUSKA reentrant pressure balances, the RUSKA 2481 oil operated and the RUSKA 2470 gas operated balances. For the oil operated balance, the distortion coefficient was found to be change from -0.7 ppm/MPa at 28 MPa to -1.96 ppm/MPa at 280 MPa (the maximum operating pressure of the balance). This reconfirmed the analysis reported by Dadson et al [1] where the equation $A = A_o(1 + \lambda P)$ does **not** hold if the cylinder of the balance is experiencing stresses in the axial direction which is the case for the RUSKA 2481 balance. For this balance, it was found that it was necessary to fit the variation (with the applied pressure) with a two-degree polynomial as: $\lambda = a + bP + cP^2$ implying a cubic term in the effective area $A = A_o(1 + aP + bP^2 + cP^3)$ where a, b and c are constants were determined {a = -0.58 ppm/MPa, b = -4.7×10^{-3} ppm/(MPa)^2 and c = 6.6×10^{-6}

ppm/(MPa)^3} with p being in MPa. Experimentally, it is difficult to obtain such a fit. If only a linear term was assumed (as is normally the case), the uncertainties would very high (of the order of 350 ppm). RUSKA have given an experimental fit for the effective area which includes terms a and b **only** {a = -1.8 ppm/MPa and b = -3.5x10^{-3} ppm/(MPa)^2, [27]} resulting in a factor of approximately two difference between simulated and experimental results for the average value of λ.

With regard to the RUSKA 2470 gas operated balance, the model developed showed an excellent agreement for value of the distortion coefficient when compared with experimental values reported by the manufacturer. The distortion coefficient was found to be essentially constant at an average value of 0.14 ppm/MPa over the (17 MPa) operating pressure range of the balance. The experimental average value of λ was reported to be approximately zero.

In conclusion, the pressure balance results show a great confidence in the models developed which feature, in summary, the following:

1. A powerful tool for the modelling of **any** pressure balance incorporating both elastic and viscosity effects.

2. Graphical representation of the pressure and gap profiles along the engagement length of a pressure balance.

3. Handling simple and complex geometries in an efficient and novel way by characterising a given geometry (using the finite element method) and thereby providing a great flexibility for further analysis including the determination of the effective area and distortion coefficient.

5. Allowance for simulation of a pressure balance outside its normal operating conditions (such as different "O" ring seal positions, different operating oil, different initial clearance gap etc) with little further expense in computing time.

All the above features are implemented in two computer programs, the **F**inite **E**lement **A**nalysis (**FEA**) program for generating base data and the **P**ressure **V**iscosity **E**lasticity (**PVE**) program for calculating (iteratively) the pressure and gap profiles. These programs are together of the order of 2500 lines of FORTRAN77 code and they run on a Sun/SPARC station.

Once the geometry for a balance has been characterised and the "base" data generated (using the FEA program), the PVE can generate pressure and gap profiles and determine the effective area and the distortion coefficient within few minutes. Therefore, further explorations (feature 5) can be simulated interactively with just a few more minutes.

The power of the models developed in this thesis for pressure balances and the availability of powerful computer workstations will lead to the possibility of developing a dedicated highly interactive package for the further understanding and future design of pressure balances, incorporating further features such as calculation of leakage flow rates, piston fall rates, simulation of the similarity method etc.

On the experimental front, there is growing interest in the measurement of pressure profiles along the engagement length of high pressure balances. The ability to simulate such profiles during the operation of the balance should lead to further work in this

area and also to lead to methods for measuring gap profiles. Since the gap and its variations are of the order of microns, it maybe possible to use optical techniques to measure gap profiles. If one of the two profiles is measured, it maybe adequate to infer the other through the simulation results. If optical techniques are used, then the developing area of optically driven silicon (Si) micromachined resonator sensors may find a place in these profile measurements.

It is hoped that in the future, simulation of other pressure balances with the fruits of this thesis work will lead to an improvement in the knowledge of the distortion coefficient for primary standard balances coupled with experimental work in laboratories maintaining National Standards. This should lead to the improvement in the agreement of inter-comparisons carried out between the National Laboratories.

REFERENCES

1. Dadson R.S., Lewis S.L., Peggs G.N., "The Pressure Balance: theory and practice", *HMSO*, London 1982.

2. Molinar G.F., Cresto P.C., Maghenzani R., " Elastic distortions in a piston-cylinder unit for pressure measurement in liquids to 100MPa: preliminary results", *Proceed. of a seminar on high pressure metrology, Pub. in BIPM monograph 89/1*, pp 12-21, Paris, May 1989.

3. Abdullah F., Finkelstein L.,"A review of mathematical modelling of instrument transducers", *Acta IMEKO IX*, pp 145-47, Budapest, Hungary, 1982.

4. Abdullah F., Mirza M.K., Rahman M.M., "Computer aided design of elastic elements in instrumentation", Triennial Int. Meas. Cong., IMEKO XI, Texas, U.S.A., Oct.16-21, 1988, *Plenaries and invited papers volume, pp 143-58, Pub. in Acta IMEKO.*

5. Turley A., "Mathematical modelling and computer aided design of diaphragms and capsules for instrument transducers", *Ph.D. thesis, City University*, London 1977.

6. Mirza M.K., "Mathematical modelling and design of snap action diaphragms", *Ph.D. thesis*, City University, London 1983.

7. Rutherford Appleton Laboratory (RAL), "The finite element library, Level 1 and Level 0", *RAL*, Chilton, Oxford, 1981, *(Now marketed by NAG Ltd., Oxford, U.K.)*.

8. Giallorenzi, T.G., Bucaro J.A., Dandridge A., Siegel Jr. G.H., Cole J.H., Rashleigh S.C., Priest R.G., "Optical fibre sensor technology", *IEEE Quantum Electron., vol. QE-1, no. 8*, pp 626-64, 1982.

9. Uttamchandani D., Thornton K.E.B., Nixon J., Culshaw B.,"Optically excited resonant diaphragm pressure sensor", *Electronic Letters, vol. 23, no. 4*, pp 152-3, Feb. 1987.

10. Grattan K.T.V., Palmer A.W., Samaan N.D., Abdullah F.,"Mathematical analysis of optically driven resonant structures in sensor applications", *IEEE J. Lightwave Tech., vol. 7, no. 1*, pp 202-08, Jan. 1989.

11. McGlade S.M., Jones G.R., "An optically powered vibrating quartz force sensor", *GEC J. Research, vol. 2*, pp 135-38, GEC Research Laboratories, Marconi Research Centre, Chelmsford, U.K., 1984.

12. Mallalieu K.I.,"Investigation into the optical activation and fibre multiplexed interrogation of vibrating sensors", *Ph.D. thesis, University of London*, U.K., 1987.

13. FEA Ltd., "LUSAS finite element analysis system", *version 86.07*, FEA Ltd., London 1986.

14. Lewis S., Peggs G.N., "The Pressure Balance: a guide to its use", *National Physical Laboratory (NPL)*, Teddington, 1979.

15. Peggs G.N., Lewis S., "The NPL primary pressure balance standard", *J. Physics E., Sci. Inst., vol. 10*, pp 1028-34, 1977.

16. BIPM, "Seminar of the CCM High Pressure Working Group at Laboratorie National d'essais", Paris, May 1988, *Pub. in BIPM monograph 89/1*, May 1989.

17. Wisniewski R., Sendys R., Wisniewski D., Rostocki A.J., "Determination of pressure coefficient of effective area of simple 0.3GPa deadweight manometer by means of finite element method of strain calculation", *Proceed. of a seminar on high pressure metrology, Pub. in BIPM monograph 89/1*, pp 27-30, Paris, May 1989.

18. Dadson R.S., Greig R.G.P., Horner A., "Developments in the accurate measurement of high pressures", *Metrologia, vol. 1*, pp 55-68, 1965.

19. Klingenberg G., "Critical remarks concerning the determination of the elastic distortion coefficient from the normal used theory", *Proceed. of a seminar on high pressure metrology, Pub. in BIPM monograph 89/1*, pp 1-11, Paris, May 1989.

20. Stuart P.R., "The influence of fluid viscosity on the distortion of piston-cylinder assemblies in pressure balances", *Proceed. of a seminar on high pressure metrology, Pub. in BIPM monograph 89/1*, pp 31-40, Paris, May 1989.

21. Legras J.C., "Piston gauge used as high accuracy standards in the range 0.01-1000MPa", *Proceed. of a seminar on high pressure metrology, Pub. in BIPM monograph 89/1*, pp 41-52, Paris, May 1989.

22. Welch B.E., Bean V.E., "Pressure and temperature measurement in the annulus between the piston and cylinder of a simple dead- weight piston gauge", *Rev. Sci. Inst., 55 (12)*, pp 1901-09, Dec. 1984.

23. Arbani I., "Finite element modelling of pressure balances", *B.Sc. final year project, Dept. of Elec. Eng., City University*, London 1989.

24. Femview Ltd., "FEMGEN/FEMVIEW user manual", *version 1*, Femview Ltd., Leicester, Oct. 1989.

25. Abdullah F., Samaan N.D., "Mathematical modelling of NPL pressure balances", *A confidential report for NPL conducted under EMRA NPL 82/0449, Meas. and Inst. Centre, City University*, London 1989.

26. National Physical Laboratory, Teddington, "*Private Communication*", London, May 1990.

27. RUSKA Inst. Corp., "*Balance operation data sheets*", RUSKA Instrument Corporation, Texas 77063, U.S.A.

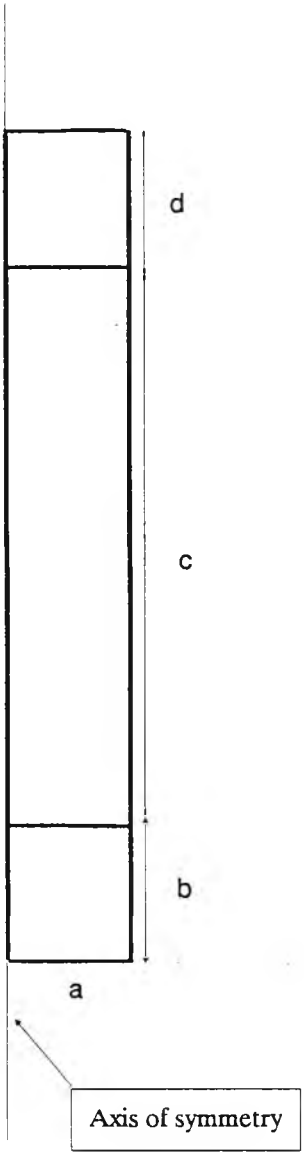
Bibliography

1. Peggs G.N. (editor), "High pressure measurement techniques", *Applied Science Publishers*, Essex, U.K., 1983.

2. Stuart P.R., "Standards for the measurement of pressure", *Meas. and Control, vol. 20,* pp 7 -11, Oct. 1987.

Appendix A

Geometrical data for NPL and RUSKA pressure balances



Piston Dimensions for the NPL Pressure Balances				
Dimension	Series			
(mm)	**100**	**200**	**300**	**400**
a	12.50	6.25	3.125	1.5625
b	5.00	5.00	5.00	5.00
c*	74.00	40.00	35.00	25.00
d	6.00	5.00	5.00	5.00
* Engagement length				

Piston Dimensions for the RUSKA 2470 Gas Operated Pressure Balance	
Dimension	*(mm)*
a	1.625
b	18.42
c*	22.23
d	10.00
* Engagement length	

Piston Dimensions for the RUSKA 2481 Oil Operated Pressure Balance	
Dimension	*(mm)*
a	1.625
b	13.17
c*	23.03
d	9.40
* Engagement length	

Cylinder Dimensions for the NPL Series 100 Pressure Balance	
Dimension	*(mm)*
a	54.00
b	32.50
c*	74.00
d	17.50
e	15.00
f	12.50

* Engagement length

Cylinder Dimensions for the NPL Series 200 to 400 Pressure Balances			
Dimension *(mm)*	Series		
	200	**300**	**400**
a	6.25	3.125	1.5625
b	13.75	16.875	18.437
c	20.00	20.00	20.00
d	10.00	10.00	10.00
e	30.00	30.00	30.00
f	20.50	23.60	25.19
g	1.40	1.24	1.24
h	2.50	2.275	2.25
i	8.60	13.76	23.76
j*	0.75	1.00	1.00
k*	40.00	35.00	25.00

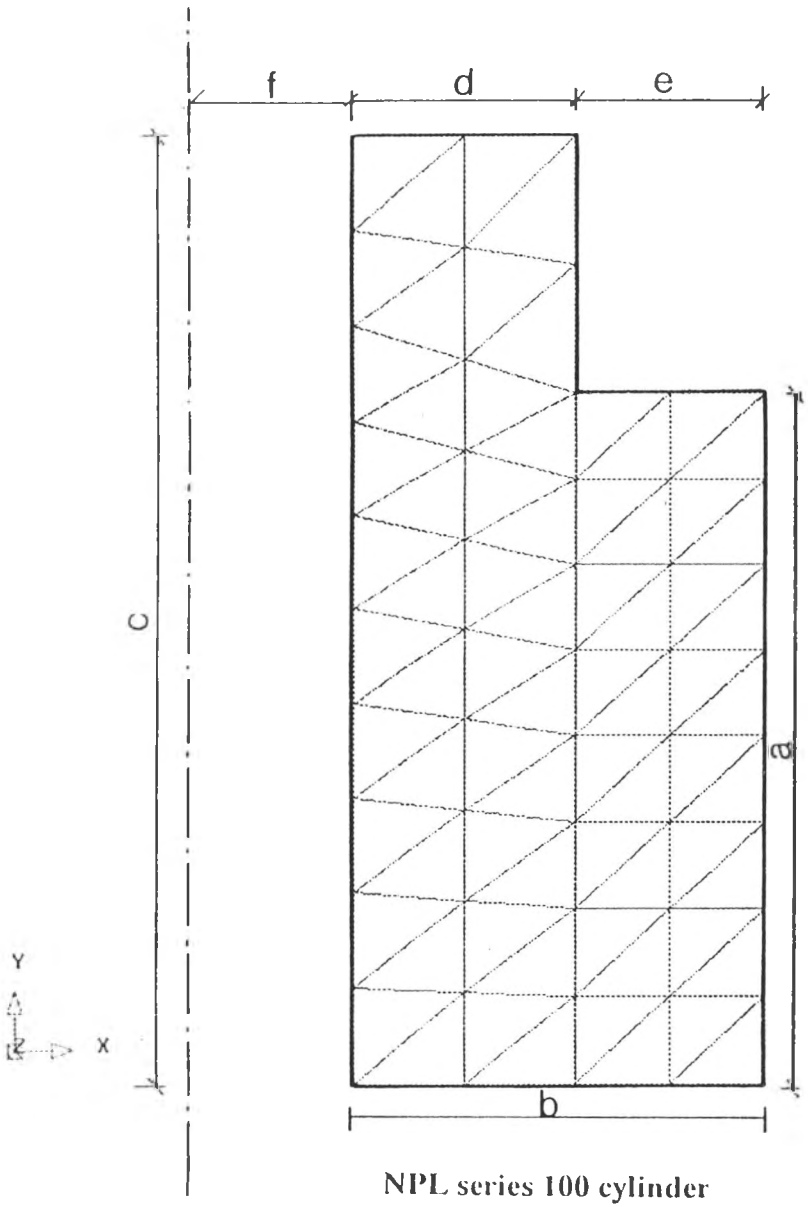
* Engagement length

Cylinder Dimensions for the RUSKA2481 Oil Operated Pressure Balance	
Dimension	*(mm)*
a	7.625
b	14.375
c	5.87
d	11.48
e	5.30
f	6.25
g	4.50
h	2.45
i	2.275
j*	1.625
k*	23.03

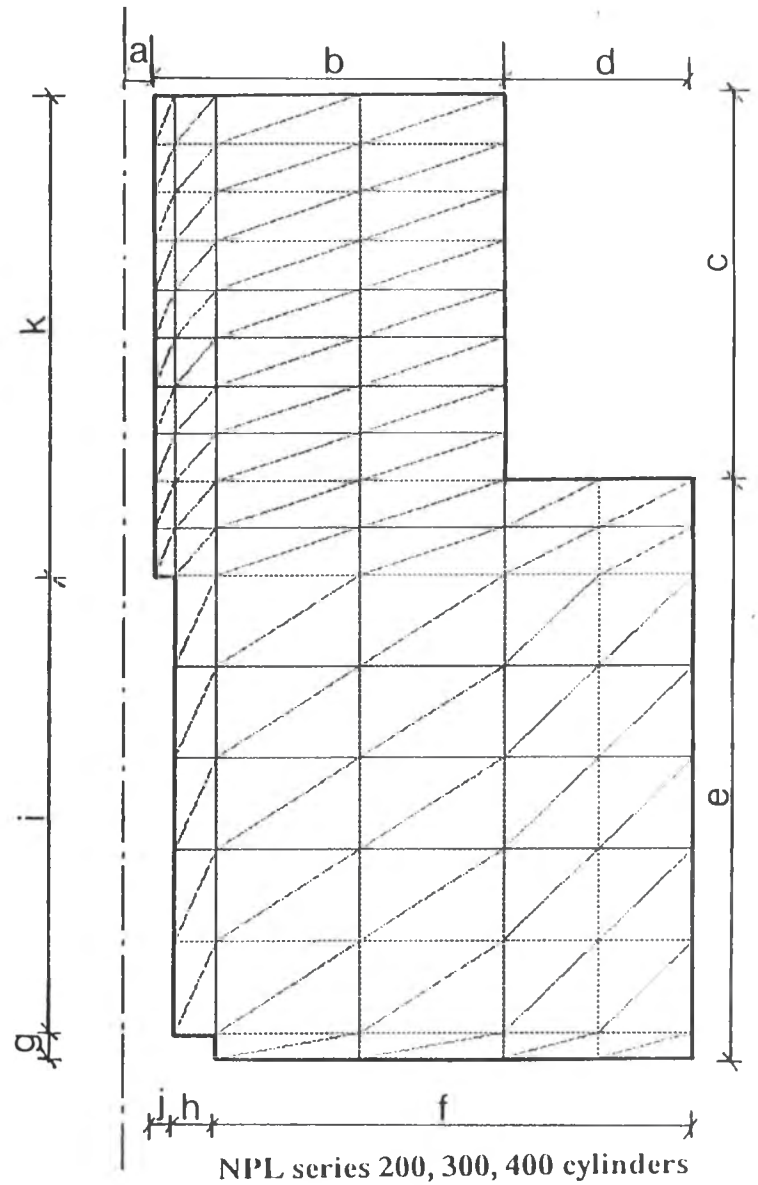
* Engagement length

Cylinder Dimensions for the RUSKA 2470 Gas Operated Pressure Balance	
Dimension	*(mm)*
a	2.285
b	3.62
c	4.75
d	10.93
e	11.30
f	10.655
g*	1.625
h*	22.23

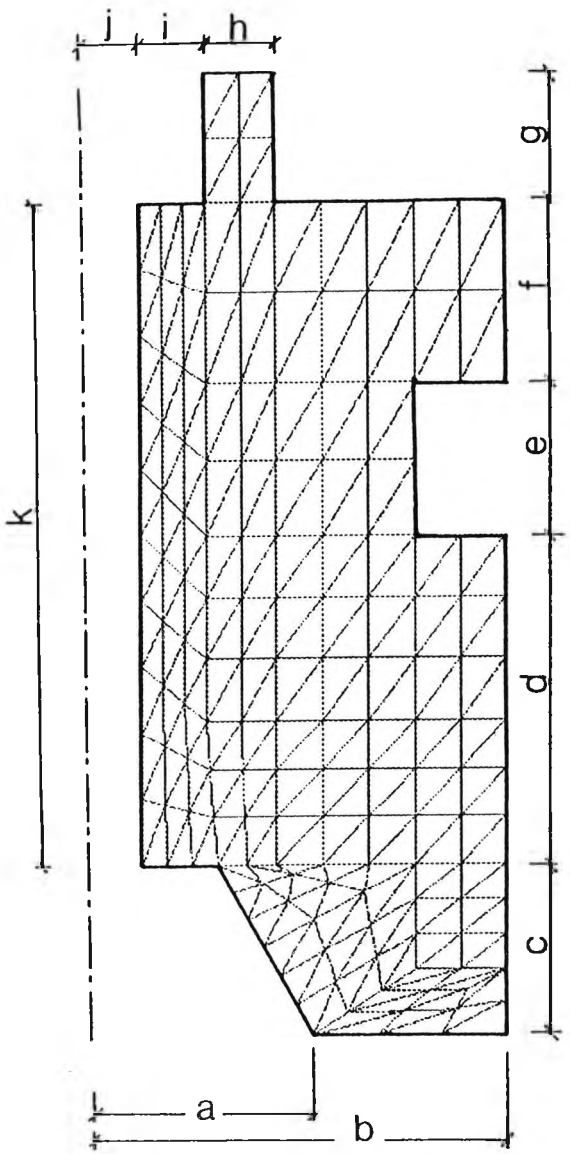
* Engagement length



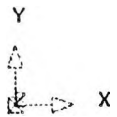
NPL series 100 cylinder

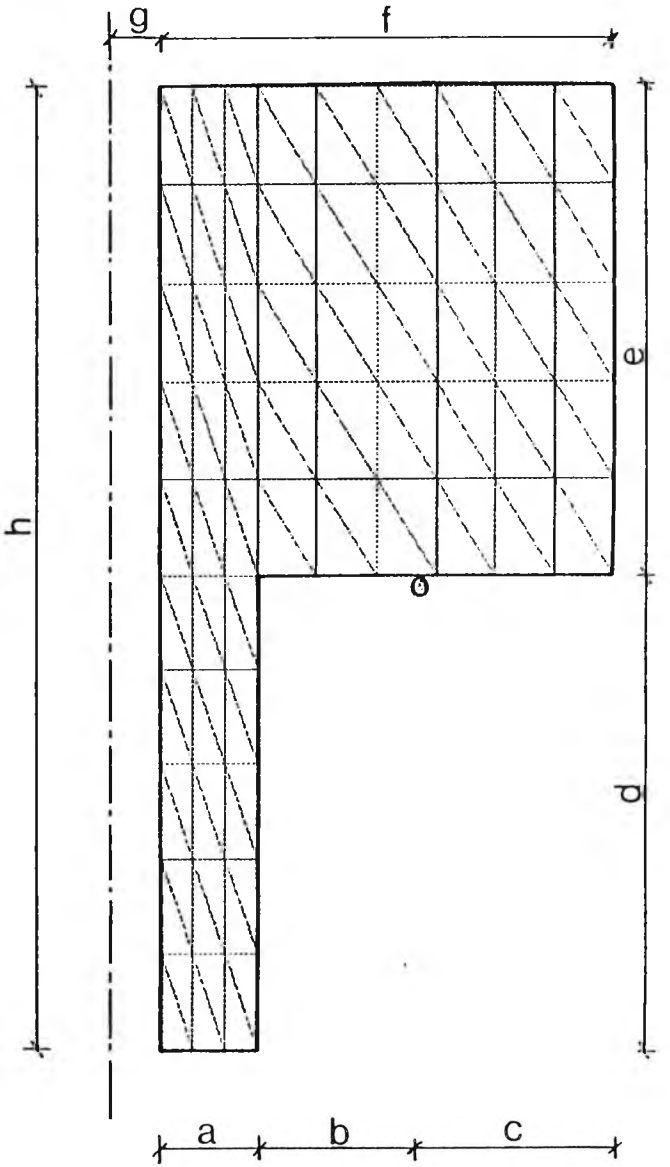


NPL series 200, 300, 400 cylinders



RUSKA 2481 cylinder





RUSKA 2470 cylinder

Appendix B

Sample Input and Output Data for the PVE Program*

* *For confidentiality reasons, a listing of the PVE program is not provided in this thesis. For availability, please contact the author.*

Data	Description
RUSKA2481/OIL	model /support condition
0	AN OIL BALANCE
10	no. of sections
p2481a	name of data file (piston) "CASE A"
c2481a	name of the data file (cylinder) "CASE A"
28.0	applied pressure (MPa)
2.0	gap (microns)
23.03	eng. length (mm)
1.625	radius (mm)
1000	no. of layers
25.8	viscosity at ambient pressure
0.67	exponent term of viscosity
1.0e-3	ERROR NORM
1.0	ATMOSPHERIC PRESSURE (in atmos)
n	no check on data file array (piston)
n	no check on data file arrays (cylinder)

Enter Pressure Balance Model & Support Condition
****
Model/Support : RUSKA2481/OIL
****

Is the balance modelled an Oil or Gas type balance?
Enter 0 if the balance being modelled is Oil type
Enter 1 if the balance being modelled is Gas type

Hint: to remember this, 0 is like the "O" in Oil

0
Enter no. of sections
10
Enter piston disp. array file name
p2481a
Enter cylinder disp. array file name
c2481a
Enter Applied Pressure in MPa
28.000000000000
Enter Initial Piston/Cylinder Gap in microns
2.0000000000000
Enter Engagement Length in mm
23.030000000000
Enter Radius of the Piston in mm
1.6250000000000
Enter Max. No. of Layers
1000
Enter Viscosity at ambient pressure. (in cP)
25.800000000000
Enter exponent term. in Visco-Press relation
0.67000000000000
Enter ERROR NORM
1.0000000000000D-03
Enter Atmospheric Pressure (in atmos)
1.0000000000000

Do you want to check any element of the array?(y,n)
n
Do you want to check any element of the array?(y,n)
n

ERROR NORM = 0.55178469028602
ERROR NORM = 1.3873361860991D-03
ERROR NORM = 8.7046630546431D-05
= =
S U M M A R Y
= =
Model/Support : RUSKA2481/OIL

Applied Pressure 28.000000000000 MPa
Initial Gap 2.0000000000000 microns
Exponent term. 0.67000000000000

Solution obtained after 3 iterations at:-
Min Gap = 2.0219595487901 microns
Max Gap = 1.9865899212426 microns
ERROR NORM of 8.7046630546431D-05
= =
****
* EFFECTIVE AREA
****
* New Area = 8.3058132048958 in sq. mm
* Old Area = 8.3059765490000 in sq. mm
* Change in Area = -19.665851839166 ppm
* Lamda = -0.70235185139878 ppm/MPa
****

Appendix C

Listing of the FEA Program*

* *For more details, please see chapter 3 section 3.11*

```
1c.....FEA.f
2c.....Finite Element Analysis (FEA)
3c234567890123456789012345678901234567890123456789012345678901234567890123456789012
4c
5       double precision abss,
6      & apress, b, bt, btdb, coord, ap, pi,
7      & d, db, det,e, elk, eta,fun,
8      & gder, geom, jac, jacin, der, loads, nu,
9      & quot, rad, radis, sysk, wght, x, xi, dsysk
10c
11      integer dif, dimen, iecnt, docnt, dofel,
12     & dofnod, elnum, eltop, eltyp, hband, iabss, ib, ibt,
13     & ibtdb, icoord, id, idb, ielk, ieltop, ifncow, ifun,
14     & igder, igeom, ijac, ijacin, ilder,
15     & iloads, inf, iquad, irestr, isteer,
16     & isysk, itest, iwght, jabss,
17     & jb, jbt, jbtdb, jcoord, jd, jdb, jelk, jeltop, jgder,
18     & jgeom, jjac, jjacin, jlder, jnf, jrestr
19c
20      integer jsysk, kn, mslop, melop,
21     & nele, nf, nin, nin1,
22     & nin2, nodel, nodnum, nout,
23     & nout1, nout2, nqp, numss, oisysk, ojsysk,
24     & resnod, restr, steer, totdof, totels, totnod, icontr
25c
26      integer ianlc, ianth,ie,iend,is,isid,iside,ist,istep,istp,
27     & lnodf5, icoln, kchkl, knoadd, noteq, lnodt,
28     & itlc, lef1, lef2, lelplc, mk, mlc, mortlc, nengn, nengn1, nengn2,
29     & nchel, ncl, nf2, nf4, nrf, nf5, ndsid,
30     & nlnods, nnps, ntle, oln, olel, orof,
31     & lcwn, mkf, mkfs, mkfe, cs, ps, chlcwn, mkn, mkns, mkfr,
32     & sumel, sumnod, num, mcowlc
33c
34      integer lef, int, iourc, i, j, k, m, engnfo, iengnfo
35c
36      integer qdrnt, iqdrnt, aqdrnt
37c
38      integer ihf,ihforc,ihf6,ihfmid,ihfoln,ivf,ivforc,ivf6,
39     & ivfmid,ivfoln,indsid,iradis,jradis,iap,iianlc,iie,iis,iisid,
40     & iistp,imortlc,ioln,izolnxc,izolnyc,iolel,iicoln,ichfoln,icvfoln,
41     & ichf,icvf,inchel,idsysk,jdsysk,iwork,ix
42c
43      real zolnxc, zolnyc, hforc, hf6, hfmid, hf, vforc, vf6,
44     & vfmid, vf, hfoln, vfoln, chf, cvf, chfoln, cvfoln,work
45c
46      parameter(ihf = 500,ihforc = 500,ihf6 = 500,ihfmid = 300, ihfoln = 500,
47     & ivf = 500 , ivforc = 500 , ivf6 = 500 , ivfmid = 300 , ivfoln = 500 ,
48     & indsid = 500 , iradis = 800,jradis = 2', iap = 300,
49     & iianlc = 300 ,iie = 300 ,iis = 300 ,iisid = 300 , iistp = 300 ,
50     & imortlc = 300 ,ioln = 500 ,izolnxc = 500 , izolnyc = 500 ,
51     & iolel = 300 , iicoln = 500 ,ichfoln = 500 , icvfoln = 500 ,
52     & ichf = 500 , icvf = 500 , inchel = 300, iengnfo = 50)
53c
54      parameter(iqdrnt = 300)
55c
56      parameter(icoord = 800,idsysk = 800,ieltop = 800,iloads = 800,inf = 800,
```

```
     & irestr = 800, isysk = 800, jcoord = 2,jdsysk = 600,jeltop = 10,jrestr = 3,
     & jsysk = 600,nnps = 3, pi = 3.1415927)
c
c
      parameter(iabss = 2, ib = 4, ibt = 12, ibtdb = 12, id = 4,
     & idb = 4,  ielk = 12, ifun = 6,
     & igder = 2,
     & igeom = 6 , ijac = 2 , ijacin = 2 , ilder = 2 ,
     & isteer = 12 ,  iwght = 7 , iwork = 3, ix = 2,
     & jabss = 7 , jb = 12 , jbt = 4 , jbtdb = 12 ,
     & jd = 4 , jdb = 12 , jelk = 12 , jgder = 6 , jgeom = 2 ,
     & jjac = 2 , jjacin = 2 , jlder = 6 , jnf = 2)
c
      dimension abss(iabss,jabss), b(ib,jb), bt(ibt,jbt),
     & btdb(ibtdb,jbtdb), d(id,jd), db(idb, jdb),
     & elk(ielk,jelk), fun(ifun),
     & gder(igder,jgder), geom(igeom,jgeom),jac(ijac,jjac),
     & jacin(ijacin, jjacin), der(ilder,jlder), steer(isteer),
     & wght(iwght), work(iwork), x(ix)
c
      dimension coord(icoord,jcoord), dsysk(idsysk,jdsysk),
     & eltop(ieltop,jeltop), loads(iloads),
     & hf(ihf),hforc(ihforc),hf6(ihf6),hfmid(ihfmid),hfoln(ihfoln),
     & vf(ivf),vforc(ivforc),vf6(ivf6),vfmid(ivfmid),vfoln(ivfoln),
     & ndsid(indsid),nf(inf,jnf), radis(iradis,jradis),
     & restr(irestr,jrestr),sysk(isysk,jsysk),ap(iap),
     & ianlc(iianlc),ie(iie),is(iis), isid(iisid), istp(iistp),
     & mortlc(imortlc), oln(ioln), zolnxc(izolnxc),zolnyc(izolnyc),
     & olel(iolel),icoln(iicoln), chfoln(ichfoln), cvfoln(icvfoln),
     & chf(ichf), cvf(icvf), nchel(inchel), engnfo(iengnfo)
c
      dimension qdrnt(iqdrnt)
c
      logical first
c
      character*6 lcfn
      character*7 lcfnd
      character*8 forfn
      character*10  fname
c
      data icontr /-99/, nin1/1/, nout1/2/, nin2 /7/, nout2 /8/,
     & ncl /9/, nrf /11/, nf2 /12/,lef1 /13/, nf4 /14/,
     & lef2 /15/, nengn1 /16/, nengn2 /17/, orof /18/, nf5 /19/,
     & mkf /30/
c
c     open input file for cylinder data generated from FEMGEN plus rest. info.
c
      open(nin1,file = 'cylina',status = 'old')
c
c     open cylinder output file for input to
c     femview conversion program
c
      open(nout1,file = 'cylictlc',status = 'unknown')
c
c     open input file for piston data generated from FEMGEN plus rest. info.
c
```

```
      open(nin2,file = 'pisina',status = 'old')
c
c      open piston output file for input to
c      femview conversion program
c
      open(nout2,file = 'pisictlc',status = 'unknown')
c
c
      open(lef1, file = 'lodcasc', status = 'old')
      open(lef2, file = 'lodcasp', status = 'old')
      open(nengn1, file = 'engnodsc', status = 'old')
      open(nengn2, file = 'engnodsp', status = 'old')
      open(nf2, file = 'lcffohvtmp', status = 'unknown')
      open(ncl,file = 'lcfchklst',status = 'unknown')
      open(nf4, file = 'lcfnff4', status = 'unknown')
      open(nf5, file = 'lcfordup', status = 'unknown')
c
      rewind (nin1)
      rewind (nout1)
      rewind (nin2)
      rewind (nout2)
c
c
      print*,'***************************************************'
      print*,'*                              *'
      print*,'*     PRESSURE   BALANCE             *'
      print*,'*                                                 *'
      print*,'*      Finite Element Program Analysis            *'
      print*,'*                                                 *'
      print*,'***************************************************'
c
c
c
      iecnt   = 2
c
      do 9999 docnt  =  1, iecnt
       if(docnt.eq.1)then
         nin = nin2
         nout = nout2
         lef  =  lef2
         nengn =  nengn2
           print*,'             = = = = = = = = = = = = = = = = = = = = = = = = = ='
           print*,'               PISTON ANALYSIS    '
           print*,'             = = = = = = = = = = = = = = = = = = = = = = = = = ='
       elseif(docnt.eq.2)then
         nin = nin1
         nout = nout1
         lef  =  lef1
         nengn =  nengn1
           print*,'             = = = = = = = = = = = = = = = = = = = = = = = = = ='
           print*,'             CYLINDER  ANALYSIS    '
           print*,'             = = = = = = = = = = = = = = = = = = = = = = = = = ='
       endif
c
      rewind (nf2)
      rewind (nf4)
```

```
      rewind (ncl)
c
c
c           ----------------------------------------------------
c           !     Set (itest) for full checking            !
c           !----------------------------------------------------
c
c
      itest = 0
      call feinit
c          .....................................................
c          .                                                   .
c          .           input nodel geometry                    .
c          .                                                   .
c          .....................................................
c
       print*,'start of inputting: nod geo., ele. top., mat prop.'
c     input of nodal geometry
c
      read(nin,2320)totnod,dimen
      do 10 i = 1,totnod
        read(nin,2330)nodnum,(coord(nodnum,j),j = 1,dimen)
        write(nout,2330)nodnum,(coord(nodnum,j),j = 1,dimen)
 10   continue
c
      call conlcf(lef, ist, iend, istep, iside, apress, mlc,
     & ianth, itlc, is, iis, ie, iie, istp, iistp, isid, iisid,
     & ap, iap, mortlc, imortlc, ianlc, iianlc,
     & nf4, lelplc, lcwn, mkf, cs, ps, chlcwn, int, lcfn,
     & mkfs, mkfe, qdrnt, iqdrnt, aqdrnt)
c
      call lechlc (mkf, chlcwn, mkn, mkns, lcwn, itlc, nf4, ist,
     & iend, istep, is, iis, ie, iie, istp, iistp, isid, iisid, lelplc,
     & mortlc, imortlc, ianlc, iianlc, ap, iap, sumel, sumnod, num,
     & nnps, lcfnd, lcfn, iside, ianth, mlc, qdrnt, iqdrnt)
c
c
c
c          .....................................................
c          .                                                   .
c          .           input of element topology               .
c          .                                                   .
c          .....................................................
c
      read(nin,2320)eltyp,totels,nodel
      do 20 i = 1,totels
        read(nin,2320)elnum,(eltop(elnum,j + 2),j = 1,nodel)
        write(nout,2320)elnum,(eltop(elnum,j + 2),j = 1,nodel)
c
      call conlcn (mkn, mkf, lcwn, chlcwn, itlc, ist, iend,
     & istep, iside, is, iis, ie, iie, istp, iistp, isid, iisid,
     & ndsid, indsid, eltop, ieltop, jeltop, ianlc, iianlc, mortlc,
     & imortlc, ntle, elnum, ap, iap, coord, icoord, jcoord, lcfn,
     & lcfnd, mkns, mlc, ianth, qdrnt, iqdrnt)
c
        eltop(elnum,1) = eltyp
```

```
225         eltop(elnum,2) = nodel
226   20   continue
227   c
228        call confor (mkn, mkfr, nf2, lcwn, chlcwn, pi, ap, iap,
229       & oln, ioln, zolnxc, izolnxc, zolnyc, izolnyc, olel, iolel, nnps,
230       & nlnods, hforc, ihforc, hf6, ihf6, hfmid, ihfmid, hf, ihf, vforc,
231       & ivforc, vf6, ivf6, vfmid, ivfmid, vf, ivf, hfoln, ihfoln, vfoln,
232       & ivfoln, lcfnd, forfn, qdrnt, iqdrnt)
233   c
234   cRelease the following STOP only to make sure that "lcfxxx" files are OK.
235   c     STOP
236   c
237   c          .....................................................
238   c          .        input of material properties and           .
239   c          .        construction of stress-strain matrix d     .
240   c          .        for plane strain                           .
241   c          .....................................................
242   c
243        read(nin,2340)nu,e
244        print*,'Poisson ratio    =', nu
245        print*,'Youngs modulus   =', e
246        call daxi (d,id,jd,e,nu,numss,itest)
247   c
248   c          .....................................................
249   c          .        input of number of degrees of freedom      .
250   c          .        per node, input of restrained node         .
251   c          .        data and construction of nodal freedom     .
252   c          .        array nf                                   .
253   c          .....................................................
254   c
255        read(nin,2320)dofnod
256        read(nin,2320)resnod
257   c    print*,' dofnod = ', dofnod
258   c    print*,' resnod = ', resnod
259        k = dofnod + 1
260   c... Free Node COUnt ... unconstrained nodes in the geometry (ifncow).
261   c
262        ifncow =  0
263        do 30 i = 1, resnod
264          read(nin,2320) (restr(i,j), j = 1, k)
265          do 35 kn = 1, k
266            if (restr(i,kn).eq.0) then
267              ifncow = ifncow + 1
268            endif
269   35     continue
270   30   continue
271   c
272        oisysk = ifncow + ((totnod - resnod) * dofnod)
273   c
274        call formnf (restr,irestr,jrestr,resnod,totnod,dofnod,nf,inf,jnf,
275       & totdof,itest)
276   c
277        print*,'nodal geo, elem. top., mat. prop: input-  ended.'
278   c
279        print*,'calc. semi-bandwidth ...'
280   c    calculation of semi-bandwidth
```

```
c
      first = .true.
      do 40 nele = 1,totels
        call fredif (nele,eltop,ieltop,jeltop,nf,inf,jnf,dofnod,first,
     & dif,itest)
 40   continue
      hband = dif + 1
c
      ojsysk = hband
c
c     print*,'oisysk = ',oisysk
c     print*,'ojsysk = ',ojsysk
c
      print*,'global stiffness matrix assembley ...'
c
c             .....................................................
c             .                                                   .
c             .       global stiffness matrix assembly            .
c             .                                                   .
c             .....................................................
c
c
      if((totdof.gt.isysk).or.(hband.gt.jsysk)) then
        print*,''
        print*,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
        print*,'INSUFFICIENT ARRAY SIZE FOR SYSTEM MATRIX'
        print*,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
        print*,''
        print*,'Execution Aborted'
        STOP
      endif
c
      call matnul (sysk,isysk,jsysk,totdof,hband,itest)
      dofel = nodel*dofnod
      call qtri7 (wght,iwght,abss,iabss,jabss,nqp,itest)
      iourc  = 0
      do 50 nele = 1,totels
        iourc  = iourc  + 1
        call elgeom (nele,eltop,ieltop,jeltop,coord,icoord,jcoord,geom,
     & igeom,jgeom,dimen,itest)
c
c     print*,'integrate elem. stiff. using nqp quad point'
c
c             .....................................................
c             .       integration loop for element stiffness      .
c             .       using nqp quadrature points                 .
c             .....................................................
c
        call matnul (elk,ielk,jelk,dofel,dofel,itest)
        do 60 iquad = 1,nqp
c
c     print*,'form lin. sh. fun., space deriv. in loc. cord'
c           .....................................................
c           .       form linear shape function and space        .
c           .       derivatives in the local corrdinates.       .
c           .       transform local derivatives to global       .
```

```
c          .          coordinate system                    .
c          ..............................................
c
      xi = abss(1,iquad)
      eta = abss(2,iquad)
      call trim6 (fun,ifun,der,ilder,jlder,xi,eta,itest)
      call matmul (der,ilder,jlder,geom,igeom,jgeom,jac,ijac,jjac,
     & dimen,nodel,dimen,itest)
      call matinv (jac,ijac,jjac,jacin,ijacin,jjacin,dimen,det,itest)
      call matmul (jacin,ijacin,jjacin,der,ilder,jlder,gder,igder,
     & jgder,dimen,dimen,nodel,itest)
c
c     print*,'formation of strain/disp. matrix'
c          ..............................................
c          .          formation of strain-displacement     .
c          .          matrix b and output to work file for .
c          .          later recovery process               .
c          ..............................................
c
c     radius at quadrature points
      rad = 0.0d0
      do 70 i = 1,nodel
        rad = rad + fun(i)*geom(i,1)
 70   continue
      call b2p2 (b,ib,jb,gder,igder,jgder,fun,ifun,geom,igeom,jgeom,
     & nodel,itest)
c
c          ..............................................
c          .          formation of element stiffness elk   .
c          ..............................................
c
c     print*,'formation of element stiffness matrix'
      call matmul (d,id,jd,b,ib,jb,db,idb,jdb,numss,numss,dofel,
     & itest)
      call matran (b,ib,jb,bt,ibt,jbt,numss,dofel,itest)
      call matmul (bt,ibt,jbt,db,idb,jdb,btdb,ibtdb,jbtdb,dofel,
     & numss,dofel,itest)
      quot = dabs(det)*wght(iquad)*2.0d0*pi*rad
      do 80 i = 1,dofel
        do 90 j = 1,dofel
          btdb(i,j) = btdb(i,j)*quot
 90     continue
 80   continue
      call matadd (elk,ielk,jelk,btdb,ibtdb,jbtdb,dofel,dofel,itest)
 60   continue
c
c          ..............................................
c          .          assembly of system stiffness matrix  .
c          .          sysk                                 .
c          ..............................................
c
c    print*,'assembley of stiffness matrix.. subs: direct,assym'
     call direct (nele,eltop,ieltop,jeltop,nf,inf,jnf,dofnod,steer,
     & isteer,itest)
      call assym (sysk,isysk,jsysk,elk,ielk,jelk,steer,isteer,hband,
     & dofel,itest)
```

```
 393   50   continue
 394   c
 395   c     print*,'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
 396   c     do 1 o1 = 1, isysk
 397   c       do 2 o2 = 1, jsysk
 398   c         if(sysk(o1,o2).ne.0.0d0) then
 399   c           print*,'sysk(',o1,',',o2,') = ', sysk(o1,o2)
 400   c         endif
 401   c2      continue
 402   c1    continue
 403   c     print*,'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
 404   c
 405   c     print*,'our check counter = ',iourc
 406   c     print*,'= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = ='
 407   c
 408   c
 409   c          .....................................................
 410   c          !                                   !
 411   c          !        Duplicate  sysk  into  dsysk          !
 412   c          !                                   !
 413   c          .....................................................
 414   c
 415         print*,'[ sysk -- dsysk ] :    duplicate started ...'
 416   c
 417   c     open(90, file = 'orsysk', status = 'unknown')
 418   c     open(91, file = 'ordsysk', status = 'unknown')
 419   c     rewind(90)
 420   c     rewind(91)
 421         do  100 i = 1, oisysk
 422           do  110 j = 1, ojsysk
 423             dsysk(i,j) = sysk(i,j)
 424   c          write(90,*) i, j, sysk(i,j)
 425   c          write(91,*) i, j, dsysk(i,j)
 426    110      continue
 427    100   continue
 428   c     close(90)
 429   c     close(91)
 430   c
 431         print*,'sysk: Duplicate completed ... dsysk formed ...'
 432   c
 433   c.... You can control the no. of load cases to be solved
 434   c.... by adjusting START and END of loop 200.
 435   c.... Default is from 1 to total no. of load cases (chlcwn)
 436   cSTART Start Load Case is mslop  (mStart Looping on Load Cases)
 437   cEND.  End   "    " is melop  (mEnd Looping on Load Cases)
 438   c
 439   c
 440         mslop   = 1
 441         melop   = chlcwn
 442         mcowlc  = 0
 443   c
 444         print*,''
 445         print*, 'Number of Load Cases to Solve for (auto set) = ', chlcwn
 446         print*,''
 447         print*,'= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = ='
 448         print*,'Solution will start from LOAD CASE NO. 1 to ', chlcwn
```

```
      print*,'IF YOU WISH TO CHANGE THIS, THEN:-'
      print*,''
      print*,' 1. Type ^C'
      print*,' 2. Copy this file to a temp file'
      print*,' 3. With the editor, LOCATE this line by:-'
      print*,'    a. type vi temp'
      print*,'    b. type :/CHANGE'
      print*,' 4. Follow instructions as advised.'
      print*,'= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = ='
      print*,''
c
c     CHANGE CHANGE  CHANGE CHANGE  CHANGE CHANGE
c
c     ************************************************************
c     * You have edited this program to change the start and END    *
c     * sequence of solving for load cases. To do that, reset      *
c     * the values of:-                                            *
c     *                                                            *
c     *    mslop  and  melop                                       *
c     *                                                            *
c     * to whatever values you wish.                               *
c     *                                                            *
c     * For example: To start from loads case no 10 and ending at 11 *
c     * change mslop to 9 and melop to 11.                         *
c     *
c     * OBVOIUSLY, you have to delete chlcwn                       *
c     *                                                            *
c     * DO THIS in the FOLLOWING 2 LINES. COMPILE and RUN          *
c     ************************************************************
c     *                                                            *
c     * PLEASE  PLEASE  PLEASE                                     *
c     * = = = = = = = = = = = = = = = = = = = = = = =                *
c     *                                                            *
c     * WHEN YOU FINISH, DELETE temp and RECOMPILE  FEA.f          *
c     *                                                            *
c     ************************************************************
c
c
      mslop  = 1
      melop  = chlcwn
c
c
      print*,'-------------------------------------------------'
      print*,'Total No. of Load Cases to be Solved = ', chlcwn
      print*,'Starting from Load Case No.        = ', mslop
      print*,'Ending at & including Load Case No.  = ', melop
      print*,'-------------------------------------------------'
c
      m     = mslop
      chlcwn  = melop
      do 200  lcwn = m , chlcwn
        print*,'Solving for Load Case No. ', lcwn
        mcowlc = mcowlc + 1
        lnodf5 = 0
        call bulcfo (lcwn, lnodf5, mkfr, nf5, oln, ioln,
     & hfoln, ihfoln, vfoln, ivfoln, forfn)
```

```
505  c
506          call mklcfr (nf5, nrf, lnodf5, kchkl, knoadd, noteq,
507       & icoln, iicoln, ncl, mk, nchel, inchel, lnodt, oln, ioln,
508       & hfoln, ihfoln, vfoln, ivfoln, chf, ichf, cvf, icvf, chfoln,
509       & ichfoln, cvfoln, icvfoln, lcwn, chlcwn, lcfnd, forfn)
510  c
511          call vecnul (loads, iloads, iloads, itest)
512          call cloads (nrf, lnodt, oln, ioln, nodnum, dofnod, nf,
513       & inf, jnf, hfoln, ihfoln, vfoln, ivfoln, work, iwork, loads,
514       & iloads)
515          call solrad (sysk, isysk, jsysk, loads, iloads, totdof,
516       & hband, itest, totnod, dofnod, nf, inf, jnf, radis, iradis,
517       & jradis, x, ix)
518          call outfv (totnod, nout, radis, iradis, jradis,
519       & dofnod, icontr)
520          call savpa (fname, orof, nengn, lef, radis, iradis,
521       & jradis, engnfo, iengnfo)
522          call chanlc (m, lcwn, mcowlc, chlcwn, docnt, iecnt)
523  c       print*,'oisysk = ', oisysk
524  c       print*,'ojsysk = ', ojsysk
525          call recovr (oisysk, ojsysk, sysk, isysk, jsysk,
526       & dsysk, idsysk, jdsysk)
527   200     continue
528        call ftsofn  (docnt, iecnt)
529  c
530  c..... nulling sysk ... ready for another geometry if any.
531  c
532        call nulsys (oisysk, ojsysk, sysk, isysk, jsysk,
533       & dsysk, idsysk, jdsysk)
534  c
535  c&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
536  c       .....................................................
537  c       .                                                   .
538  c       .                  F o r m a t s                    .
539  c       .                                                   .
540  c       .              from  1200  to  2900                 .
541  c       .             9010 & 9020 are reserved              .
542  c       .          commented formats act as an index        .
543  c       .....................................................
544  c
545  c1250  format(2x, i5, 3(8x,e14.7))
546  c1290  format(e14.7)
547  c1300  format(i5)
548   2320  format(16i5)
549   2330  format(i5,2f10.5)
550   2340  format(2f10.0)
551  c2400  format(i5, 2(2x,f10.5), i5, f10.3, 2x, i5)
552  c2410  format(i5, 2x, e14.7, 2x, e14.7)
553  c2420  format(e14.7, 2x, e14.7)
554  c2430  format(f10.0)
555  c2440  format(a10)
556  c2450  format('lcf', I3.3)
557  c2460  format(4(2x, i5), 2x, f10.3, 3(2x, i5))
558  c2470  format('lcfn', I3.3)
559  c2480  format('lcfor', I3.3)
560  c
```

```
561  c
562        call fefin
563  c
5649999 continue
565  c
566        close (nin1)
567        close (nin2)
568        close (nout1)
569        close (nout2)
570        close (ncl)
571        close (nf2)
572        close (nf4)
573        close (nf5)
574        close (nrf)
575        close (lef1)
576        close (lef2)
577        close (nengn1)
578        close (nengn2)
579  c
580  c
581  c
582        stop
583        end
584  c
585  c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
586  c          .....................................................
587  c          .                                                   .
588  c          .    S U B R O U T I N E S                          .
589  c          .    ******************************                 .
590  c          .                                                   .
591  c          .....................................................
592  c
593  c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
594  c...  sub. conlcf. Breaks the I/P load case file into individual files.
595  c
596        subroutine conlcf(lef, ist, iend, istep, iside, apress, mlc,
597       & ianth, itlc, is, iis, ie, iie, istp, iistp, isid, iisid,
598       & ap, iap, mortlc, imortlc, ianlc, iianlc,
599       & nf4, lelplc, lcwn, mkf, cs, ps, chlcwn, int, lcfn,
600       & mkfs, mkfe, qdrnt, iqdrnt, aqdrnt)
601  c
602        integer lef, ist, iend, istep, iside, mlc, ianth, itlc,
603       & nf4, lelplc, lcwn, mkf, cs, ps, chlcwn, is, ie, istp,
604       & mkfs, mkfe, int, isid, mortlc, ianlc, i, j, k
605  c
606        integer iis, iie, iistp, iisid, imortlc, iianlc, iap
607  c
608        integer qdrnt, iqdrnt, aqdrnt
609  c
610        double precision apress, ap(iap)
611        dimension is(iis), ie(iie), istp(iistp),isid(iisid),
612       & mortlc(imortlc),  ianlc(iianlc)
613  c
614        dimension qdrnt(iqdrnt)
615  c
616        character*6 lcfn
```

```
      rewind(lef)
      itlc = 0
      mkfs = mkf
c     print*,'in conlcf... mkf = ',mkf
 3000 read (lef, 1300) ist, iend, istep, iside
      read (lef, 2430) apress
c     print*,'apress =', apress
      read (lef, 1300) aqdrnt, mlc, ianth
      itlc = itlc + 1
        if ((ianth.eq.0).and.(mlc.eq.0)) then
          go to 3010
        else
          go to 3000
        endif
c
 3010 print*,'constructing load case files...'
c3010 print*,'total no. of sub load cases = ',itlc
c
      rewind (lef)
      do 3020 i = 1, itlc
        read (lef, 1300) is(i), ie(i), istp(i), isid(i)
c       print*,'is(',i,') =',is(i)
c       print*,'ie(',i,') =',ie(i)
c       print*,'istp(',i,') =',istp(i)
c       print*,'isid(',i,') =',isid(i)
        read (lef, 2430) ap(i)
c       print*,'ap(',i,') =',ap(i)
        read (lef, 1300) qdrnt(i), mortlc(i), ianlc(i)
c       print*,'mortlc(',i,') =',mortlc(i)
c       print*,'ianlc(',i,') =',ianlc(i)
 3020 continue
c
      lcwn = 0
      do 3050 i = 1, itlc
        if((mortlc(i).eq.0.and.ianlc(i).eq.1).or.
     & (mortlc(i).eq.0.and.ianlc(i).eq.0)) then
          lcwn = lcwn + 1
        endif
 3050 continue
c
      chlcwn = lcwn
      lcwn = 0
      rewind (lef)
      do 3060 i = 1, itlc
        read (lef, 1300) is(i), ie(i), istp(i), isid(i)
        read (lef, 2430) ap(i)
        read (lef, 1300) qdrnt(i), mortlc(i), ianlc(i)
        j = mortlc(i)
        k = ianlc(i)
        if (i.eq.1) then
          lcwn = lcwn + 1
          call mklcf (lcwn, int, lcfn)
          open (mkf, file = lcfn, status = 'unknown')
          rewind (mkf)
          write (mkf, 2460) is(i), ie(i), istp(i), isid(i),
     & ap(i), qdrnt(i), mortlc(i), ianlc(i)
```

```
673             if (j.eq.0.and.k.eq.0) then
674               if(i.eq.itlc) then
675                 go to 3060
676               else
677                 print*,'ERROR1: LOAD CASE FILE FORMAT FAILED'
678                 print*,'Execution Aborted...'
679                 STOP
680               endif
681             endif
682             go to 3060
683           endif
684           if(j.eq.0.and.k.eq.0) then
685             cs = 0
686           elseif (j.eq.0.and.k.eq.1) then
687             cs = 1
688           elseif (j.eq.1.and.k.eq.0) then
689             cs = 2
690           elseif (j.eq.1.and.k.eq.1) then
691             cs = 3
692           endif
693   c
694           if (mortlc(i-1).eq.0.and.ianlc(i-1).eq.0) then
695             ps = 0
696           elseif (mortlc(i-1).eq.0.and.ianlc(i-1).eq.1) then
697             ps = 1
698           elseif (mortlc(i-1).eq.1.and.ianlc(i-1).eq.0) then
699             ps = 2
700           elseif (mortlc(i-1).eq.1.and.ianlc(i-1).eq.1) then
701             ps = 3
702           endif
703   c
704           if ((cs.eq.0).or.(cs.eq.2)) then
705             if (ps.eq.0) then
706               print*,'ERROR2: TOO MANY (END) IN LOAD CASE FILE'
707               print*,'Execution Aborted...'
708               STOP
709             elseif (ps.eq.1) then
710               lcwn = lcwn + 1
711               call mklcf (lcwn, int, lcfn)
712               mkf = mkf + 1
713   c.................. print*,'cs02/ps1... mkf = ', mkf
714               open (mkf, file = lcfn, status = 'unknown')
715               rewind (mkf)
716               write (mkf, 2460) is(i), ie(i), istp(i), isid(i),
717       & ap(i), qdrnt(i), mortlc(i), ianlc(i)
718               go to 3060
719             elseif (ps.eq.2) then
720               write (mkf, 2460) is(i), ie(i), istp(i), isid(i),
721       & ap(i), qdrnt(i), mortlc(i), ianlc(i)
722               go to 3060
723             elseif (ps.eq.3) then
724               print*,'ERROR3: LOAD CASE FILE SEQUENCE MISMATCH'
725               print*,'Execution Aborted...'
726               STOP
727             endif
728           endif
```

```
729  c
730          if ((cs.eq.1).or.(cs.eq.3)) then
731            if (ps.eq.0) then
732              print*,'ERROR4: TOO MANY (END) IN LOAD CASE FILE'
733              print*,'Execution Aborted...'
734              STOP
735            elseif (ps.eq.1) then
736              lcwn = lcwn + 1
737              call mklcf (lcwn, int, lcfn)
738              mkf = mkf + 1
739  c.....  print*,'cs13/ps1... mkf = ',mkf
740              open (mkf, file = lcfn, status = 'unknown')
741              rewind (mkf)
742              write (mkf, 2460) is(i), ie(i), istp(i), isid(i),
743       & ap(i), qdrnt(i), mortlc(i), ianlc(i)
744              go to 3060
745            elseif (ps.eq.2) then
746              print*,'ERROR5: LOAD CASE FILE SEQUENCE MISMATCH'
747              print*,'Execution Aborted...'
748              STOP
749            elseif (ps.eq.3) then
750              write (mkf, 2460) is(i), ie(i), istp(i), isid(i),
751       & ap(i), qdrnt(i), mortlc(i), ianlc(i)
752              go to 3060
753            endif
754          endif
755  c
756   3060  continue
757        if (lcwn.ne.chlcwn) then
758          print*,'ERROR6: CONSTRUCTED LOAD CASE FILES VS I/P FILE'
759          print*,'      Check:  FAILED'
760          print*,'Execution Aborted...'
761          STOP
762        endif
763        mkfe = mkfs + chlcwn - 1
764        do 3070 i = mkfs, mkfe
765          close (i)
766   3070  continue
767        mkf = mkfs
768   1300  format(i5)
769   2430  format(f10.0)
770   2460  format(4(2x, i5), 2x, f10.3, 3(2x, i5))
771  c     print*,'out of conlcf.. mkf =',mkf
772        return
773        end
774  c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
775  c...  sub. mklcf. Creates load cse files to accomodate input load cases.
776  c
777        subroutine mklcf (lcwn, int, lcfn)
778        integer lcwn, int
779        character*6 lcfn
780        int = lcwn
781        write(lcfn, 2450) int
782   2450  format('lcf', I3.3)
783        return
784        end
```

```
785  c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
786  c...  sub. lechlc. Checks no. of elem. per load case. Get no. of nodes.
787  c...  Write them in their approp. files lcfns (Load Case Files incl. Nodes).
788  c
789        subroutine lechlc (mkf, chlcwn, mkn, mkns, lcwn, itlc, nf4, ist,
790       & iend, istep, is, iis, ie, iie, istp, iistp, isid, iisid, lelplc,
791       & mortlc, imortlc, ianlc, iianlc, ap, iap, sumel, sumnod, num,
792       & nnps, lcfnd, lcfn, iside, ianth, mlc, qdrnt, iqdrnt)
793  c
794        integer mkf, mkn, lcwn, chlcwn, itlc, nf4, ist, iend, istep,
795       & nnps, lelplc, sumel, sumnod, num, mkns, i, m, is, ie, istp,
796       & iside, ianth, mlc, isid, mortlc, ianlc
797  c
798        integer iis, iie, iistp, iisid, imortlc, iianlc, iap
799  c
800        integer qdrnt, iqdrnt
801  c
802        double precision ap(iap)
803        dimension is(iis), ie(iie), istp(iistp),isid(iisid),
804       & mortlc(imortlc), ianlc(iianlc)
805  c
806        dimension qdrnt(iqdrnt)
807  c
808        character*6 lcfn
809        character*7 lcfnd
810        mkn = mkf + 1
811        mkns = mkn
812        do 24000 lcwn = 1, chlcwn
813          mkn = mkn + 1
814          call mklfcn (lcwn, lcfnd)
815          open (mkn, file = lcfnd, status = 'unknown')
816          rewind (mkn)
817  c
818          call detlc (mkf, itlc, ist, iend, istep, iside, ianth,
819       & mlc, is, iis, ie, iie, isid, iisid, istp, iistp, mortlc,
820       & imortlc, ianlc, iianlc, ap, iap, lcwn, lcfn, qdrnt, iqdrnt)
821  c
822          if (lcwn.gt.1) then
823  c        print*,'in lechlc .. after call detlcdetlc.. nf4 = ', nf4
824          endif
825          rewind (nf4)
826          do 24005 i = 1, itlc
827            ist    = is(i)
828            iend   = ie(i)
829            istep  = istp(i)
830            lelplc = 0
831            do 24010 m = ist, iend, istep
832              lelplc = lelplc + 1
833  24010       continue
834            write (nf4, 1300) lelplc
835  24005     continue
836          rewind (nf4)
837          sumel = 0
838          do 24015 i = 1, itlc
839            read (nf4, 1300) num
840            sumel = sumel + num
```

```
 841 24015    continue
 842          sumnod = sumel * nnps
 843          write (mkn, 1300) sumnod
 844 24000 continue
 845 c     print*,'after 24000.. mkn =',mkn
 846       mkn = mkns
 847 c     print*,' out of lechlc .. mkn  =', mkn
 848 c     print*,'mkns =', mkns
 849 1300  format (i5)
 850       return
 851       end
 852 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
 853 c...  sub. detlc. Gets the details of all parameters in each load case from
 854 c...  lcfs files. Counts itlc for each.
 855 c
 856 c
 857       subroutine detlc (mkf, itlc, ist, iend, istep, iside, ianth,
 858      & mlc, is, iis, ie, iie, isid, iisid, istp, iistp, mortlc,
 859      & imortlc, ianlc, iianlc, ap, iap, lcwn, lcfn, qdrnt, iqdrnt)
 860 c
 861       integer lcwn, int, mkf, itlc, ist, iend, istep, iside, ianth,
 862      & mlc, is, ie, isid, istp, mortlc, ianlc, i
 863 c
 864       integer iis, iie, iistp, iisid, imortlc, iianlc, iap
 865 c
 866       integer qdrnt, iqdrnt, aqdrnt
 867 c
 868       double precision ap(iap), apress
 869       dimension is(iis), ie(iie), istp(iistp),isid(iisid),
 870      & mortlc(imortlc), ianlc(iianlc)
 871 c
 872       dimension qdrnt(iqdrnt)
 873 c
 874       character*6 lcfn
 875       itlc = 0
 876 c     print*,'in  detlc .. mkf =', mkf
 877       call mklcf (lcwn, int, lcfn)
 878       open (mkf, file = lcfn, status = 'unknown')
 879       rewind (mkf)
 880 25000 read (mkf, 2460) ist, iend, istep, iside, apress, aqdrnt,
 881      & mlc, ianth
 882       itlc = itlc + 1
 883       if (mlc.eq.0) then
 884          go to 25005
 885       else
 886          go to 25000
 887       endif
 888 c
 889 25005 rewind (mkf)
 890       do 25010 i = 1, itlc
 891          read (mkf, 2460) is(i), ie(i), istp(i), isid(i), ap(i),
 892      & qdrnt(i), mortlc(i), ianlc(i)
 893 25010 continue
 894       close (mkf)
 895  2460 format(4(2x, i5), 2x, f10.3, 3(2x, i5))
 896       return
```

```
      end
c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
c...  sub. mklfcn. Generates load case files (same no. as lcfs). These
c...  will include total loaded nodes per load case, nodes, coord, elem no.
c...  applied pressure.
c
      subroutine mklfcn (lcwn, lcfnd)
      integer lcwn
      character*7 lcfnd
      write (lcfnd, 2470) lcwn
 2470 format ('lcfn', I3.3)
      return
      end
c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
c...  subroutine to construct the Load Case Nodes files for each load case.
c...  It should also include total no. of loaded nodes at very top, (loca. 1).
c...  This is done in sub. lechlc. When you get here, all the lcfs
c...  are already open & the pointer is at location 2 of the file.
c
      subroutine conlcn (mkn, mkf, lcwn, chlcwn, itlc, ist, iend,
     & istep, iside, is, iis, ie, iie, istp, iistp, isid, iisid,
     & ndsid, indsid, eltop, ieltop, jeltop, ianlc, iianlc, mortlc,
     & imortlc, ntle, elnum, ap, iap, coord, icoord, jcoord, lcfn,
     & lcfnd, mkns, mlc, ianth, qdrnt, iqdrnt)
c
      integer mkn, mkf, lcwn, chlcwn, itlc, ist, iend, istep, iside,
     & ntle, elnum, mkns, m, j, k, mlc, ianth,is, ie, istp, isid,
     & ndsid, eltop, ieltop, jeltop, icoord, jcoord, mortlc,
     & ianlc, inel
c
      integer iis, iie, iistp, iisid, indsid, imortlc, iianlc,
     & ianth, mlc, iap
c
      integer qdrnt, iqdrnt
c
      double precision coord (icoord,jcoord), ap(iap)
      dimension is(iis),ie(iie),istp(iistp),isid(iisid),
     & ndsid(indsid), mortlc(imortlc), ianlc(iianlc),
     & eltop(ieltop,jeltop)
c
      dimension qdrnt(iqdrnt)
c
      character*6 lcfn
      character*7 lcfnd
      mkns = mkn
c     print*,' just before 4000 ...mkn =',mkn
      do 4000 lcwn = 1, chlcwn
        mkn = mkn +1
c       if (lcwn.gt.1) then
c       print*,'in conlc/ mkf used in detlc.within 4000/mkf = ',mkf
c       print*,'within 4000.. mkn = ',mkn
c       endif
c
c
        call detlc (mkf, itlc, ist, iend, istep, iside, ianth,
     & mlc, is, iis, ie, iie, isid, iisid, istp, iistp, mortlc,
```

```
     & imortlc, ianlc, iianlc, ap, iap, lcwn, lcfn, qdrnt, iqdrnt)
c
        call mklfcn (lcwn, lcfnd)
        do 4005 m = 1, itlc
          ist   = is(m)
          iend  = ie(m)
          istep = istp(m)
          iside = isid(m)
          ntle  = 0
c         print*,'elnum =',elnum
c
          do 4010 inel = ist, iend, istep
c           print*,'inel = ',inel
            if(elnum.eq.inel) then
              ntle = ntle + 1
              if(iside.eq.1) then
                do 4020 j = 1, 3
                  ndsid(j) = eltop(elnum, j+2)
                write (mkn, 2400) ndsid(j), coord(ndsid(j),1),
     & coord(ndsid(j),2), elnum, ap(m), qdrnt(m)
c               print*,'writting into nf1(sid1) elnum=',elnum
c             print*,'writting into nf1(sid1) ndsid=',ndsid(j)
 4020             continue
c
              elseif(iside.eq.2) then
                do 4030 j = 1, 3
                  ndsid(j) = eltop(elnum, j+4)
                write (mkn, 2400) ndsid(j), coord(ndsid(j),1),
     &coord(ndsid(j),2), elnum, ap(m), qdrnt(m)
c               print*,'writting into nf1(sid2) elnum=',elnum
c             print*,'writting into nf1(sid2) ndsid=',ndsid(j)
 4030             continue
c
              elseif(iside.eq.3) then
                do 4040 k = 1, 2
                  ndsid(k) = eltop(elnum, k+6)
                write (mkn, 2400) ndsid(k), coord(ndsid(k),1),
     &coord(ndsid(k),2), elnum, ap(m), qdrnt(m)
c               print*,'writting into nf1(sid3) elnum=',elnum
c             print*,'writting into nf1(sid3) ndsid=',ndsid(k)
 4040             continue
                ndsid(3) = eltop(elnum,3)
              write (mkn, 2400) ndsid(3), coord(ndsid(3),1),
     &coord(ndsid(3),2), elnum, ap(m), qdrnt(m)
c               print*,'writting into nf1(sid3) elnum=',elnum
c             print*,'writting into nf1(sid3) ndsid=',ndsid(3)
              endif
            endif
 4010       continue
 4005     continue
 4000 continue
      mkn = mkns
c...  never close mkn's here.!!. this sub. is called for each elenum
c...  read from pisina file without calling lechlc sub. again
c...  where mkn's are originally opened
c     mkne = mkns + 1 + chlcwn
```

```
1009 c     do 4060 i = mkns, mkne
1010 c       close(i)
1011 c4060 continue
1012  2400 format(i5, 2(2x,f10.5), i5, f10.3, 2x, i5)
1013       return
1014       end
1015 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1016 c...  construct files nf2 and lcforXXX.nf2 contains the all horiz. and
1017 c...  vertical forces of all nodes.lcforXXX contains same as nf2 BUT sorted
1018 c...  as for each node appropriately with node numbers associated.
1019 c
1020       subroutine confor (mkn, mkfr, nf2, lcwn, chlcwn, pi, ap, iap,
1021      & oln, ioln, zolnxc, izolnxc, zolnyc, izolnyc, olel, iolel, nnps,
1022      & nlnods, hforc, ihforc, hf6, ihf6, hfmid, ihfmid, hf, ihf, vforc,
1023      & ivforc, vf6, ivf6, vfmid, ivfmid, vf, ivf, hfoln, ihfoln, vfoln,
1024      & ivfoln, lcfnd, forfn, qdrnt, iqdrnt)
1025 c
1026       integer mkn, mkns, mkfr, lcwn, chlcwn, nf2, oln,
1027      & olel, nnps, nlnods, mkne, i
1028 c
1029       real zolnxc, zolnyc, hforc, hf6, hfmid, hf, vforc, vf6,
1030      & vfmid, vf, hfoln, vfoln, fdeex, fdeey, radius
1031 c
1032 c     real fdee
1033 c
1034       integer izolnxc, izolnyc, ihf, ihforc, ihf6, ihfmid,
1035      & ivf, ivforc, ivf6, ivfmid, ihfoln, ivfoln, iolel, ioln,
1036      & iap
1037 c
1038       integer qdrnt, iqdrnt
1039 c
1040       double precision ap(iap), pi
1041       dimension zolnxc(izolnxc), zolnyc(izolnyc),
1042      & hf(ihf), hforc(ihforc), hf6(ihf6), hfmid(ihfmid),
1043      & vf(ivf), vforc(ivforc), vf6(ivf6), vfmid(ivfmid),
1044      & hfoln(ihfoln), vfoln(ivfoln),olel(iolel),oln(ioln)
1045 c
1046       real fds, sinth, costh
1047       dimension qdrnt(iqdrnt)
1048 c
1049       character*7 lcfnd
1050       character*8 forfn
1051       mkns = mkn
1052       mkne = mkns + 1 + chlcwn
1053 c
1054 c     print*,'in confor/ closing all mkn(s)'
1055 c     print*,'mkn = ',mkn
1056 c     print*,'mkns  = ',mkns
1057 c     print*,'mkne  = ',mkne
1058 c     print*,'chlcwn = ',chlcwn
1059 c
1060       do 4999 i = mkns, mkne
1061         close(i)
1062  4999 continue
1063       mkn  = mkns
1064       mkfr = mkn + 1
```

```
1065 c
1066 c     print*,'out of 4999/into 5000/.. mkn = ',mkn
1067 c     print*,'mkfr = ',mkfr
1068 c     print*,'nf2 = ',nf2
1069 c     print*,'this routine operates only on two channels at a time'
1070 c     print*,'these are mkn & mkfr/ they are closed each time'
1071 c
1072       do 5000 lcwn = 1, chlcwn
1073         call mklfcn (lcwn, lcfnd)
1074         call mkforf (lcwn, forfn)
1075         open (mkn, file = lcfnd, status = 'unknown')
1076         open (mkfr, file = forfn, status = 'unknown')
1077         rewind (mkn)
1078         rewind (mkfr)
1079         rewind (nf2)
1080         read (mkn, 1300) nlnods
1081         write (mkfr, 1300) nlnods
1082         do 5040 i = 1, nlnods
1083           read (mkn, 2400) oln(i), zolnxc(i), zolnyc(i), olel(i),
1084     & ap(i), qdrnt(i)
1085  5040     continue
1086 c
1087         do 5050 i = 1, nlnods, nnps
1088           fdeex  = abs (zolnxc(i) - zolnxc(i+2))
1089           fdeey  = abs (zolnyc(i) - zolnyc(i+2))
1090           radius = zolnxc(i+1)
1091 c          print*,'radius = ',radius
1092 c
1093 c..... for vertical nodes where x coord are same
1094 c..... radius is always the xc of the midside node
1095 cxx           if (fdeex.lt.1.0d-10) then
1096 cxx             fdee  = fdeey
1097 cxx             if((qdrnt(i).eq.1).or.(qdrnt(i).eq.4)) then
1098 cxx               ap(i) = ap(i)
1099 cxxc              print*,'vertical element ... +ve pressure'
1100 cxxc              print*,'qdrnt(',i,') = ', qdrnt(i)
1101 cxxc              print*,'ap(',',',i,') = ', ap(i)
1102 cxx             elseif((qdrnt(i).eq.2).or.(qdrnt(i).eq.3)) then
1103 cxx               ap(i) = -1.0d0 * ap(i)
1104 cxxc              print*,'vertical element ... -ve pressure'
1105 cxxc              print*,'qdrnt(',i,') = ', qdrnt(i)
1106 cxxc              print*,'ap(',i,') = ', ap(i)
1107 cxx             endif
1108 cxx             hforc(i)   = 2.0d0*pi*radius*fdee*ap(i)
1109 cxx             hf6(i)     = hforc(i)/6.0
1110 cxx             hfmid(i)   = 4.0 * hf6(i)
1111 cxx             vf(i)      = 0.0
1112 cxx             write (nf2, 2420) hf6(i), vf(i)
1113 cxx             write (nf2, 2420) hfmid(i), vf(i)
1114 cxx             write (nf2, 2420) hf6(i), vf(i)
1115 cxx           endif
1116 cxxc..... for horiz. nodes where y coord are same
1117 cxxc..... radius is always the xc of the midside node
1118 cxx           if (fdeey.lt.1.0d-10) then
1119 cxx             fdee  = fdeex
1120 cxx             if((qdrnt(i).eq.1).or.(qdrnt(i).eq.2)) then
```

```
1121 cxx              ap(i) = ap(i)
1122 cxxc             print*,'horizontal element ... +ve pressure'
1123 cxxc             print*,'qdrnt(',i,') = ', qdrnt(i)
1124 cxxc             print*,'ap(',',',i,') = ', ap(i)
1125 cxx            elseif((qdrnt(i).eq.3).or.(qdrnt(i).eq.4)) then
1126 cxx              ap(i) = -1.0d0 * ap(i)
1127 cxxc             print*,'horizontal element ... -ve pressure'
1128 cxxc             print*,'qdrnt(',i,') = ', qdrnt(i)
1129 cxxc             print*,'ap(',i,') = ', ap(i)
1130 cxx            endif
1131 cxx            vforc(i)   = 2.0d0*pi*radius*fdee*ap(i)
1132 cxx            vf6(i)     = vforc(i)/6.0d0
1133 cxx            vfmid(i)   = 4.0d0 * vf6(i)
1134 cxx            hf(i)      = 0.0
1135 cxx            write (nf2, 2420) hf(i), vf6(i)
1136 cxx            write (nf2, 2420) hf(i), vfmid(i)
1137 cxx            write (nf2, 2420) hf(i), vf6(i)
1138 cxx          endif
1139 cxxc
1140 cxx          if((fdeex.gt.1.0d-10).and.(fdeey.gt.1.0d-10)) then
1141              fds       = sqrt(fdeex**2 + fdeey**2)
1142              sinth     = fdeey/fds
1143              costh     = fdeex/fds
1144              hforc(i)  = 2*pi*radius*fds*ap(i)*sinth
1145              vforc(i)  = 2*pi*radius*fds*ap(i)*costh
1146              hf6(i)    = hforc(i)/6.0
1147              hfmid(i)  = 4.0 * hf6(i)
1148              vf6(i)    = vforc(i)/6.0d0
1149              vfmid(i)  = 4.0d0 * vf6(i)
1150              if(qdrnt(i).eq.1) then
1151                write (nf2, 2420) hf6(i), vf6(i)
1152                write (nf2, 2420) hfmid(i), vfmid(i)
1153                write (nf2, 2420) hf6(i), vf6(i)
1154              elseif(qdrnt(i).eq.2) then
1155                hf6(i)  = -1.0d0 * hf6(i)
1156                hfmid(i) = -1.0d0 * hfmid(i)
1157                write (nf2, 2420) hf6(i), vf6(i)
1158                write (nf2, 2420) hfmid(i), vfmid(i)
1159                write (nf2, 2420) hf6(i), vf6(i)
1160              elseif(qdrnt(i).eq.3) then
1161                hf6(i)  = -1.0d0 * hf6(i)
1162                hfmid(i) = -1.0d0 * hfmid(i)
1163                vf6(i)  = -1.0d0 * vf6(i)
1164                vfmid(i) = -1.0d0 * vfmid(i)
1165                write (nf2, 2420) hf6(i), vf6(i)
1166                write (nf2, 2420) hfmid(i), vfmid(i)
1167                write (nf2, 2420) hf6(i), vf6(i)
1168              elseif(qdrnt(i).eq.4) then
1169                vf6(i)  = -1.0d0 * vf6(i)
1170                vfmid(i) = -1.0d0 * vfmid(i)
1171                write (nf2, 2420) hf6(i), vf6(i)
1172                write (nf2, 2420) hfmid(i), vfmid(i)
1173                write (nf2, 2420) hf6(i), vf6(i)
1174              else
1175                print*,'ERROR100: SUB. (confor) FAILED TO DETECT'
1176                print*,'       AN ELEMENT AT A SLOPE.'
```

```
1177                print*,'Execution Aborted'
1178                STOP
1179              endif
1180 cxx          endif
1181 c
1182  5050    continue
1183 c
1184      rewind (nf2)
1185      do 5070 i = 1, nlnods
1186        read (nf2, 2420) hfoln(i), vfoln (i)
1187        write(mkfr, 2410) oln(i), hfoln(i), vfoln (i)
1188  5070 continue
1189      close (mkn)
1190      close (mkfr)
1191  5000 continue
1192 c
1193  1300 format(i5)
1194  2400 format(i5, 2(2x,f10.5), i5, f10.3, 2x, i5)
1195  2410 format(i5, 2x, e14.7, 2x, e14.7)
1196  2420 format(e14.7, 2x, e14.7)
1197      return
1198      end
1199 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1200 c...  sub mkforf. Creats lcforXXX file names. Used in confor sub.
1201 c
1202      subroutine mkforf (lcwn, forfn)
1203      integer lcwn
1204      character*8 forfn
1205      write (forfn, 2480) lcwn
1206  2480 format ('lcfor', I3.3)
1207      return
1208      end
1209 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1210 c...  sub. (mklcfr) constructs  file which includes the combined
1211 c...  identical loaded nodes. It operates on the f5 for one complete
1212 c...  load case just before solving.
1213 c
1214      subroutine mklcfr (nf5, nrf, lnodf5, kchkl, knoadd, noteq,
1215     & icoln, iicoln, ncl, mk, nchel, inchel, lnodt, oln, ioln,
1216     & hfoln, ihfoln, vfoln, ivfoln, chf, ichf, cvf, icvf, chfoln,
1217     & ichfoln, cvfoln, icvfoln, lcwn, chlcwn, lcfnd, forfn)
1218 c
1219      integer nf5, nrf, lnodf5, kchkl, knoadd, noteq,icoln,
1220     & lcwn, chlcwn, i, j,
1221     & ncl, mk, nchel, lnodt, oln
1222 c
1223      integer ihfoln, ivfoln, ichf, icvf, ichfoln, icvfoln, iicoln,
1224     & inchel, ioln
1225 c
1226      real hfoln, vfoln, chf, cvf, chfoln, cvfoln
1227      dimension hfoln(ihfoln), vfoln(ivfoln),chf(ichf),cvf(icvf),
1228     & chfoln(ichfoln), cvfoln(icvfoln),icoln(iicoln),nchel(inchel),
1229     & oln(ioln)
1230      character*7 lcfnd
1231      character*8 forfn
1232 c...  nrf is closed in (cloads) sub. ....
```

```
1233 c
1234       open (nrf, file = 'lcfored', status = 'unknown')
1235       rewind (nrf)
1236       rewind (nf5)
1237 c..... print*,'lnodf5 = ',lnodf5
1238       do 6010 i = 1, lnodf5
1239         read (nf5, 2410) oln(i), hfoln(i), vfoln(i)
1240  6010  continue
1241 c
1242       rewind (nf5)
1243       rewind (ncl)
1244       kchkl  = 0
1245       knoadd = 0
1246       noteq  = 0
1247 c
1248       do 6020 j = 1, lnodf5
1249 c       print*,'in 7020 &&&&&&&&&&&&&&&'
1250         read (nf5, 2410) icoln(j), chfoln(j), cvfoln(j)
1251 c       print*,'icoln(',j,') = ',icoln(j)
1252 c       print*,'chfoln(',j,') = ',chfoln(j)
1253 c       print*,'cvfoln(',j,') = ',cvfoln(j)
1254         chf(j) = chfoln(j)
1255         cvf(j) = cvfoln(j)
1256 c       print*,'before if kchkl .ne. 0 ..'
1257 c       print*,'kchkl = ',kchkl
1258         if (kchkl.ne.0) then
1259           rewind (ncl)
1260           do 6030 mk = 1, kchkl
1261 c           print*,'I am in 7030 ##############'
1262             read (ncl, 1300) nchel(mk)
1263 c           print*,'nchel(',mk,') =', nchel(mk)
1264 c           print*,'icoln(',j,') =', icoln(j)
1265             if (icoln(j).eq.nchel(mk)) then
1266 c           print*,'I am within if icoln = nchel- I am going to 7020'
1267               go to 6020
1268             endif
1269  6030       continue
1270         endif
1271 c
1272         if (j.eq.lnodf5) then
1273           write (nrf, 2410) icoln(j), chf(j), cvf(j)
1274           noteq = noteq + 1
1275           go to 6020
1276         endif
1277 c
1278         do 6040 i = j+1, lnodf5
1279 c         print*,'in 6040 .... '
1280           if (icoln(j).eq.oln(i)) then
1281 c           print*,'before addition within 6040'
1282 c           print*,'chf(',j,') =',chf(j)
1283 c           print*,'cvf(',j,') =',cvf(j)
1284             chf(j) = chf(j) + hfoln(i)
1285             cvf(j) = cvf(j) + vfoln(i)
1286 c           print*,'after addition within 6040'
1287 c           print*,'chf(',j,') =',chf(j)
1288 c           print*,'cvf(',j,') =',cvf(j)
```

```
1289              knoadd = knoadd + 1
1290 c            print*,'knoadd = ',knoadd
1291 c            print*,'before if knoadd.eq.1 ...'
1292 c            print*,'kchkl = ',kchkl
1293              if (knoadd.eq.1) then
1294                kchkl = kchkl + 1
1295                write (ncl, 1300) icoln(j)
1296 c               print*,'into chklist ',icoln(j)
1297              endif
1298            endif
1299  6040      continue
1300 c
1301 c         print*,'Just after  6040 ..before if knoadd.eq.0.'
1302 c         print*,'noteq =',noteq
1303           if (knoadd.eq.0) then
1304             noteq = noteq + 1
1305           endif
1306 c         print*,'Just after 6040 .. after if knoadd.eq.0.'
1307 c         print*,'noteq =',noteq
1308 c
1309           knoadd = 0
1310 c         print*,'**************** into rf3  *****************'
1311           write (nrf, 2410) icoln(j), chf(j), cvf(j)
1312 c         print 2410, icoln(j), chf(j), cvf(j)
1313 c
1314  6020  continue
1315 c     print*,'kchkl = ',kchkl
1316 c     print*,'noteq = ',noteq
1317       lnodt = noteq + kchkl
1318 c     print*,'lnodt =',lnodt
1319       print*,'load vector assembly file for this load case ...'
1320       print*,'file: lcfored  ... ... constructed ...'
1321  1300  format (i5)
1322  2410  format(i5, 2x, e14.7, 2x, e14.7)
1323       return
1324       end
1325 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1326 c...  sub. bulcfo . Back Up Load Case FOrce file.
1327       subroutine bulcfo (lcwn, lnodf5, mkfr, nf5, oln, ioln,
1328      & hfoln, ihfoln, vfoln, ivfoln, forfn)
1329       integer lcwn, lnodf5, mkfr, nf5, oln, i
1330 c
1331       integer ihfoln, ivfoln, ioln
1332 c
1333       real hfoln, vfoln
1334       dimension hfoln(ihfoln), vfoln(ivfoln), oln(ioln)
1335       character*8 forfn
1336 c..... print*,'in bulcfo/ mkfr is only used. op&closed/ mkfr =',mkfr
1337       call mkforf (lcwn, forfn)
1338       open (mkfr, file = forfn, status = 'unknown')
1339       rewind (mkfr)
1340       rewind (nf5)
1341       read (mkfr, 1300) lnodf5
1342       do 6000 i = 1, lnodf5
1343          read (mkfr, 2410) oln(i), hfoln(i), vfoln(i)
1344          write(nf5, 2410) oln(i), hfoln(i), vfoln(i)
```

```
 6000 continue
      close (mkfr)
 1300 format (i5)
 2410 format(i5, 2x, e14.7, 2x, e14.7)
      return
      end
c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
c..... subroutine cloads .. constructs the loads array after reducing lcforXXX
c..... This operates on lcfored file which has total no. of"lines" = lnodt,(from
c..... mklcfr subroutine.
c..... (call vecnul) is to be called every time needed out of this routine
c
      subroutine cloads (nrf, lnodt, oln, ioln, nodnum, dofnod, nf,
     & inf, jnf, hfoln, ihfoln, vfoln, ivfoln, work, iwork, loads,
     & iloads)
      integer nrf, lnodt, oln, nodnum, dofnod, nf, i, j, k
      integer ihfoln, ivfoln, iwork,ioln, inf, jnf, iloads
      real hfoln, vfoln, work
c
      double precision loads (iloads)
      dimension hfoln(ihfoln), vfoln(ivfoln), work(iwork),oln(ioln),
     & nf(inf,jnf)
c
      print*,'no. of loaded nodes for this load case =', lnodt
c
c     print*,'in cloads.. nrf is only used/ nrf = ', nrf
      rewind (nrf)
      do 7000 i = 1, lnodt
c       print*,'into 7000 ----------'
        read (nrf, 2410) oln(i), hfoln(i), vfoln(i)
        nodnum = oln(i)
        do 7010 j = 1, dofnod
          if (j.eq.1) then
            work(j) = hfoln(i)
          endif
          if (j.eq.2) then
            work(j) = vfoln(i)
          endif
 7010     continue
c
        do 7020 j = 1, dofnod
          k = nf(nodnum, j)
          if (k.eq.0) then
            go to 7020
          endif
          loads(k) = work(j)
 7020     continue
 7000 continue
c
c     print*,'out of 7000 . & closing nrf ...'
C.....CLOSING NRF unit no.
      close (nrf)
 2410 format(i5, 2x, e14.7, 2x, e14.7)
      return
      end
c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

```
1401 c.... subroutine for solution and generating (radis) .
1402 c
1403       subroutine solrad (sysk, isysk, jsysk, loads, iloads, totdof,
1404      & hband, itest, totnod, dofnod, nf, inf, jnf, radis, iradis,
1405      & jradis, x, ix)
1406       double precision sysk(isysk,jsysk),radis(iradis,jradis),x(ix),
1407      & loads (iloads)
1408       dimension nf(inf, jnf)
1409       integer isysk, jsysk, iloads, totdof, hband, itest, totnod,
1410      & dofnod, nf, i, k, inf, jnf, ix, iradis, jradis
1411 c
1412 c          print*,'.....'
1413 c          print*,'subroutine CHOSOL called'
1414           print*,'equation solution in progress - please wait...'
1415 c          print*,'.....'
1416         call chosol (sysk,isysk,jsysk,loads,iloads,totdof,hband,itest)
1417 c          print*,'return from CHOSOL .....'
1418           print*,'solution : done ... ... ...'
1419         do 8000 i = 1,totnod
1420           do 8010 k = 1,dofnod
1421             if(nf(i,k).ne.0)then
1422               x(k) = loads(nf(i,k))
1423               radis(i,k) = loads(nf(i,k))
1424             else
1425               x(k) = 0.0d0
1426               radis(i,k) = 0.0d0
1427             endif
1428  8010        continue
1429  8000      continue
1430 c
1431 c.....
1432 c       do 8020 i = 1, totnod
1433 c         do 8030 k = 1, dofnod
1434 c           if (nf(i,k).ne.0) then
1435 c             print*,'nf(',i,',',k,') = ',nf(i,k)
1436 c             print*,'loads(',nf(i,k),') = ',loads(nf(i,k))
1437 c           endif
1438 c8030      continue
1439 c8020    continue
1440 c.....
1441 c       do 8040 k = 1, dofnod
1442 c         do 8050 i = 1, totnod
1443 c           print*,'displacement (',i,',',k,' )  is :'
1444 c           print 1290,radis(i,k)
1445 c8050      continue
1446 c8040    continue
1447 c
1448       return
1449       end
1450 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1451 c...  subroutine for writting output results into (ic) file for FEMVIEW
1452 c
1453       subroutine outfv (totnod, nout, radis, iradis, jradis,
1454      & dofnod, icontr)
1455       double precision radis (iradis, jradis)
1456       integer totnod, nout, dofnod, icontr, i, k, iradis, jradis
```

```
1457 c
1458       do 8060 i = 1, totnod
1459          write (nout, 1250)i,(radis(i,k),k=1,dofnod)
1460  8060  continue
1461       write (nout, 1300) icontr
1462  1250  format(2x, i5, 3(8x,e14.7))
1463  1300  format (i5)
1464       return
1465       end
1466 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1467 c... subroutine for saving results for post analysis of pressure profile
1468 c
1469       subroutine savpa (fname, orof, nengn, lef, radis, iradis,
1470      & jradis, engnfo, iengnfo)
1471       integer engnfo, iengnfo, orof, nengn, lef, i, iradis, jradis,
1472      & totengn
1473 c
1474       double precision radis (iradis, jradis)
1475       dimension engnfo(iengnfo)
1476       character*10 fname
1477       read (lef, 2440) fname
1478       open (orof, file=fname, status='unknown')
1479       print*,'results into load case file ... file name: ', fname
1480       rewind (orof)
1481       rewind (nengn)
1482       read (nengn,1300) totengn
1483       do 99000 i = 1, totengn
1484          read (nengn,1300) engnfo(i)
1485  99000  continue
1486       do 99010 i= 1, totengn
1487          write(orof, 1290) radis(engnfo(i),1)
1488 c        print*,'writting into fname .....'
1489  99010  continue
1490       close (orof)
1491  1290  format (e14.7)
1492  1300  format (i5)
1493  2440  format (a10)
1494       return
1495       end
1496 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1497 c...  subroutine for checking all load case are done. Verify itlc vs m.
1498 c
1499       subroutine chanlc (m, lcwn, mcowlc, chlcwn, docnt, iecnt)
1500 c
1501       integer m, lcwn, mcowlc, chlcwn, docnt, iecnt, j
1502 c
1503       print*,'check for another load case ...'
1504 c     print*,'mcowlc = ',mcowlc
1505       j  = chlcwn - m + 1
1506       if (mcowlc.eq.j) then
1507            if ((docnt.eq.1).and.(docnt.eq.iecnt)) then
1508               print*,'****************************************'
1509               print*,'No. of Loops on CHOSOL ...  Check: PASS'
1510               print*,'End of Piston Analysis ...'
1511               print*,'****************************************'
1512               print*,'End of Analysis ... ... AUTO STOP : PASS'
```

```
1513              print*,'****************************************'
1514  c... Next STOP is AUTO .. NEVER DELETE ..
1515                 STOP
1516              elseif (docnt.eq.1) then
1517                 print*,'****************************************'
1518                 print*,'No. of Loops on CHOSOL ...  Check: PASS'
1519                 print*,'End of Piston Analysis ...'
1520                 print*,'****************************************'
1521                 go to 9020
1522              elseif (docnt.eq.iecnt) then
1523                 print*,'****************************************'
1524                 print*,'No. of Loops on CHOSOL ...  Check: PASS'
1525                 print*,'End of Cylinder Analysis ...'
1526                 print*,'****************************************'
1527                 print*,'End of Analysis ... ... AUTO STOP : PASS'
1528                 print*,'****************************************'
1529  c... Next STOP is AUTO .. NEVER DELETE ..
1530                 STOP
1531              endif
1532           print*,'ERROR7: Number of Loops on CHOSOL ...  Check: FAILED'
1533           print*,'Execution Aborted...'
1534           print*,'****************************************************'
1535  c... Never Delete the next STOP
1536           STOP
1537        endif
1538   9020 return
1539        end
1540  c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1541  c...  subroutine for recovery of sysk from dsysk before next solution.
1542  c
1543        subroutine recovr (oisysk, ojsysk, sysk, isysk, jsysk,
1544       & dsysk, idsysk, jdsysk)
1545        integer oisysk, ojsysk, l, lm, isysk, jsysk, idsysk, jdsysk
1546        double precision sysk(isysk, jsysk), dsysk(idsysk, jdsysk)
1547  c
1548  c     open(93, file = 'recovsysk', status = 'unknown')
1549  c     open(91, file = 'orsysk', status = 'old')
1550  c     rewind(93)
1551  c     rewind(91)
1552        print*,'sysk :  start recovery  .... '
1553        do 9030 l = 1, oisysk
1554           do 9040 lm = 1, ojsysk
1555              sysk(l,lm) = dsysk(l,lm)
1556  c           write(93,*) l, lm, sysk(l,lm)
1557  c           read(91,*) l, lm, sysk(l,lm)
1558   9040     continue
1559   9030  continue
1560  c     close(93)
1561  c     close(91)
1562        print*,'sysk :  end  recovery  .... '
1563  c
1564        return
1565        end
1566  c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1567  c...  subroutine to force the program to "STOP", should the (chanlc) sub.
1568  c...  failes to stop or acknowledge. (i.e. slips)
```

```
1569 c     (Force To Stop On FiNish)
1570       subroutine ftsofn  (docnt, iecnt)
1571       integer docnt, iecnt
1572       if (docnt.eq.iecnt) then
1573            print*,'ERROR8: ANALYSIS PROGRAM FAILED TO AUTO STOP.'
1574            print*,'*****************************************'
1575            print*,'SORRY: Subroutine ftsofn ACTIVATED MANUAL STOP.'
1576            print*,'= = = = = = = = = = = = = = = = = = = = = = = ='
1577            STOP
1578       endif
1579       return
1580       end
1581 c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
1582 c...  subroutine for nulling  sysk and  dsysk before next geometry if any.
1583 c
1584       subroutine nulsys (oisysk, ojsysk, sysk, isysk, jsysk,
1585      & dsysk, idsysk, jdsysk)
1586       integer oisysk, ojsysk, null, nullm, isysk, jsysk, idsysk, jdsysk
1587       double precision sysk (isysk, jsysk), dsysk (idsysk, jdsysk)
1588c
1589       print*,'sysk, dsysk :  null start     .... '
1590       do 29030 null = 1, oisysk
1591          do 29040 nullm = 1, ojsysk
1592             sysk(null,nullm)  = 0.0d0
1593             dsysk(null,nullm) = 0.0d0
159429040     continue
159529030  continue
1596       print*,'sysk, dsysk :  null end       .... '
1597c
1598       return
1599       end
1600c= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

Appendix D

A Publication by the Author

Mathematical Analysis of Optically Powered Quartz Resonant Structures in Sensor Applications

K. T. V. GRATTAN, A. W. PALMER, N. D. SAMAAN, AND F. ABDULLAH

Abstract—A mathematical analysis of quartz resonant structures used in fiber optic sensor systems and driven optically is undertaken using analytical and finite element methods. A study was made simulating photothermal excitation of the structure to study important parameters relating to heat flow, the input optical duty cycle, the maximum displacement and the frequency-load relationship, and comparisons made with the results of experimental investigations. The extension of such work to other materials used in resonant sensor devices is discussed.

I. Introduction

A. General Introduction

INTENSIVE research work over the last several years has resulted in the publication of many schemes for the sensing of important physical and chemical parameters by optical means, using fiber optic systems. Such work has been reviewed by a number of authors [1]–[3] and as the technology has begun to mature, the directions for future research are becoming more clear. For example, considerable interest has been shown in systems which do provide an output in a nonanalog form and the transition of this new technology into the working environment (with the necessary retraining of staff involved) may be eased by the use of sensor concepts and practices familiar from electrical and electronic sensor schemes to produce so-called "hybrid" fiber optic sensors. Hence the specific advantages of optical sensors in terms of their immunity to interference, inherent safety, etc., can more clearly be appreciated, and their adoption could become more widespread.

B. Quartz Resonators

The piezoelectric quartz resonator has been used as a frequency standard for many years and the frequency sensitivity to strain/stress in quartz oscillators has been used as the basis of a number of conventional sensors, for force and pressure measurement in particular. Quartz has a high Q factor, is readily available due to its use in conventional sensors and in addition it has good optical properties. It is very stable, can sustain both compressive and tensile stresses, and has a low aging rate. Thus is it not surprising that optically based (in addition to conventional electrically driven) quartz sensors have been developed and described in the literature for the measurement of force and pressure [4]–[12]. Quartz crystal resonators have the potential to be used in any sensor scheme where the parameter to be measured can be converted to a stress effect in the crystal and thus to a corresponding resonant frequency change. As a result, there is a need to investigate these devices fully to extract their potential for optical sensor applications, to benefit from their digital form of (frequency) information output.

C. Requirements

In spite of the usefulness of the quartz element as a sensor, as yet little mathematical analysis of the performance and potential of such devices in sensor applications has been undertaken. Some authors [6]–[8] have emphasized, in their analyses, the resonant frequency and mode shapes of simple structures in quartz based sensors for different uses. In this paper, the more detailed analytical techniques recently made available (including the use of the finite element method) are implemented to model an optically driven quartz crystal plate, typical of that used in the fiber optic sensor schemes discussed previously by some of the authors [10] and other workers [2], [9], to yield a fuller assessment of various physical aspects of a prototype sensor scheme. In such an analysis, it is considered important to use a practical and experimentally demonstrated example as the basis of the verification of the model and thus make comparison with actual results of an experiment. Thus it could then be used more generally and widely for its principle objective, the design of resonator sensor systems without recourse to detailed experimental investigation of preprototype devices. Additionally, the modeling is not limited only to one material, such as quartz, and the characteristics of other important resonant sensor materials, e.g., silicon [19] can equally be predicted. Thus this leads to the development of a valuable tool for the sensor designer.

II. Physical Design Considerations

A. Theoretical Background—Elementary Considerations

In this modeling exercise the sensors of interest are those based on a vibrating quartz crystal executing a flexural mode of vibration, such as the practical systems re-

Manuscript received December 22, 1987.

This work of one author was supported by SRC/Space and Astronomy Research Center, Baghdad, Iraq.

The authors are with the Measurement and Instrumentation Center, Department of Electrical, Electronic, and Information Engineering, City University, London EC1V 0HB, England.

IEEE Log Number 8821138.

0733-8724/89/0100-0202$01.00 © 1989 IEEE

cently realized and reported in literature [4], [9], [12]. A schematic diagram of the crystal to be modeled and typical of such sensor elements is shown in Fig. 1. Dimensions used are those of McGlade *et al.* and Mallalieu [4], [13], where available, which are similar to the sensor element size in the work of Grattan *et al.* [10]. It is assumed that the quartz crystal is clamped at both ends and it is driven optically by light from an input fiber. Thus the quartz crystal vibrates due to the photothermal effect [9], as it is coated with a thin layer of a material which absorbs the incident optical power. The material is chosen to be absorbant at a wavelength of the light source (most conveniently a LED) causing the resonance in the crystal, where the modulation frequency is the resonant frequency of the quartz crystal. The absorption of the light over the period that the source is energized heats the crystal and thus an ac stress component is applied in addition to the dc stress which is present due to the axial load generated by the measured parameter. Usually the latter effect is enlarged by the special mechanical support used which is designed as appropriate for a particular specific experimental application. A continuous stream of pulses will cause the crystal to vibrate at its resonant frequency, which is primarily a function of its physical parameters and geometry. The resonant frequency is given by [12]:

$$f_0 = 1.028\,[t/l^2][E/\rho]^{1/2} \qquad (1)$$

where f_0 is the resonant frequency, t is the thickness, l is the length of quartz crystal, E is the Young's Modulus, and ρ is the mass density. The effect of the axial force on the resonant frequency is well known and quoted in the literature [6]–[8]. The frequency of the fundamental f when a dc stress is applied along the length of the crystal is given by [12]:

$$f^2 = f_0^2 + [\delta/\rho\Lambda^2] \qquad (2)$$

where the effective length l, $\Lambda/l = (4/3)$, and δ is the tensile stress applied along the length of the quartz crystal.

B. Discussion of Previous Work

A simple theoretical analysis of the motion, resonant frequency, and mode shapes of a quartz crystal clamped at both ends and vibrating in the flexural mode has previously been reported [6], [7]. A modified approach is mentioned by Albert [6], which uses a variation of the Rayleigh Method to determine the resonant fequency and the mode shapes, assuming simple harmonic motion. The results obtained were in agreement with those published by Paros [8] taking the same equation describing the motion of the quartz crystal experiencing vibration in the flexural mode. On the other hand, other systems have been implemented to measure the resultant force on the basis of driving a quartz crystal optically and electrically, and agreement was found with the frequency/load relation published by Mallalieu [13]. In the work of these authors, the photothermal effect has been analyzed in more detail to include the temperature profile through the quartz and

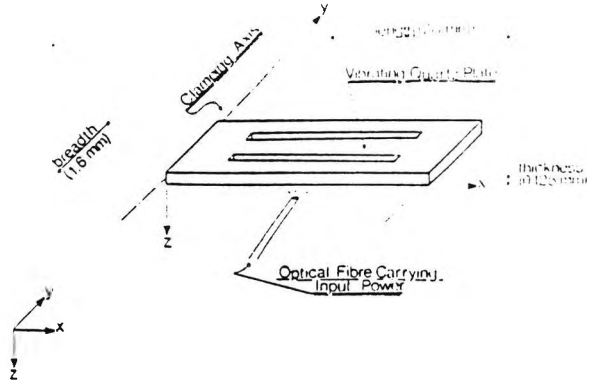


Fig. 1. Schematic diagram of the quartz plate element and optical fiber providing incident energy.

the nichrome coating which acts as the optical absorber. A brief analysis is also shown in the work of Dieulesaint *et al.* [11] with some practical considerations. McGlade *et al.* and Mallalieu [4], [13] also show the frequency/load response of the optically driven crystal vibrating under the same conditions as mentioned above. Grattan *et al.* [10], [14] discuss an experimental optically driven resonator system where the resonant frequency is determined by the coupling of light through the crystal from a critically cut fiber just below the crystal by means of the evanescent wave emerging from that fiber. The analysis to be discussed has implications for this type of experimental system.

C. Aims and Objectives of This Work

The objectives of this work described in this paper are as follows.

The development of a computer program simulating the photothermal conversion due to the absorption of optical energy in order to study the sensitivity of the process, in addition to studying the heat flow (temperature profile) through the thickness of the quartz crystal, by means of the finite element method.

The study of the effect of duty cycle of the input signal (a modulated optical signal) on a prescribed displacement, assuming that the average power of the input signal is fixed. The practical consequences of such conditions are also investigated.

The investigation of the resonant frequency of the quartz structure as well as the frequency-load relationship, both determined using the finite element method. Thus, mode shapes can be deduced and in particular the first mode of vibration.

The theoretical determination of the displacement of the driven quartz for a constant optical power input under both static and dynamic conditions using two simple analytical methods in addition to the finite element approach.

D. Development of the Model

A block diagram of the energy conversion process in such sensors is shown in Fig. 2. The optical input power is modulated via the appropriate circuitry at a frequency which is the resonant frequency of the quartz structure.

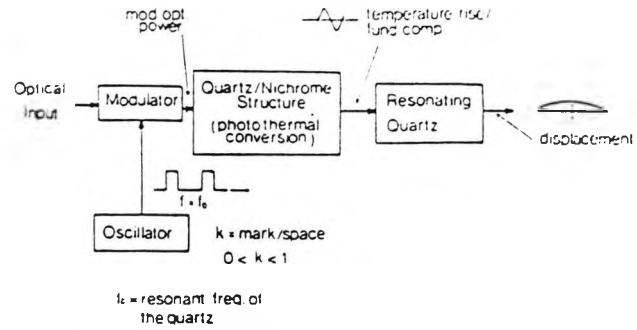


Fig. 2. Block diagram of the energy conversion process.

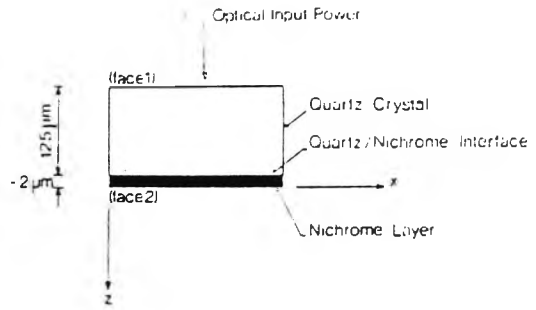


Fig. 3. Schematic diagram of the cross section through the quartz/nichrome structure.

This modulated output is incident on the quartz which is coated by an absorbing nichrome layer, yielding a temperature rise at the quartz/nichrome interface causing the crystal to deflect. The form of the temperature rise is the fundamental component of the input modulated power.

The schematic diagram illustrating a section through the sensor is shown in Fig. 3. The modulated optical power travels through the thickness of the quartz with no disturbance until it reaches the nichrome layer at the quartz/nichrome interface, where it is assumed all the optical power is absorbed. The nichrome layer has to be thick enough in order to absorb all the optical power, yet thin enough to ensure a time constant less than the period of the modulated input signal. No heat is assumed to travel back, away from the quartz/nichrome interface and temperatures at both face 1 and face 2 are assumed to be at the reference room temperature. The bottom face is assumed to be an infinite heat sink.

The dimensions of the quartz crystal studied are as shown on Fig. 1. Being typical of crystals used in recent experimental studies, they yield resonant frequencies in the tens of kilohertz range, which are easily measured optoelectronically. The x-axis is defined as being along the length of the quartz, the y-axis along the breadth of the quartz and the z-axis (thickness) is perpendicular to the paper (out of plane). The clamping axes are along the y-axis in the xy-plane at both ends.

The objectives described require a detailed review of the work of Mallalieu *et al.* [9] by solving Fourier heat equations governing the photothermal conversion, finite element analysis of the heat flow through the quartz thickness and the implementation of Fourier analysis applied to a pulse from a source of fixed average input power. The latter involves the Fourier analysis of a pulse train input, as used in the experiment of McGlade *et al.* [4]. As the fundamental component of this signal is driving the quartz at its resonant frequency, then at a constant average input power, the magnitude of the fundamental component of input power is given by:

$$y = (P \sin(k\pi))/k\pi \tag{3}$$

where k is the duty cycle of the modulated input signal, and P is the average input power in milliwatts.

The first analytical method used to determine the maximum displacement is to approximate the quartz to a simple beam, and a uniform temperature increase will cause the quartz to increase its length [15] by Δl, which is given by:

$$\Delta l = \alpha(\Delta T)l \tag{4}$$

where α is the linear thermal expansion coefficient, ΔT is the increase in temperature, and l is the original length of the bar. As shown in Fig. 4(a), the linear temperature differential between the top and bottom surface of the quartz results a curvature of the beam, which gives a lateral deflection. The change in temperature, in this case due to the absorption of optical power, causes a rotation in the top and bottom sides of the quartz with respect to each other through an angle $d\theta$ where:

$$\kappa = \frac{d\theta}{dx} = -[\alpha(T_2 - T_1)/t]. \tag{5}$$

Further, T_1 and T_2 are the top and bottom surface temperatures, t is the thickness of the quartz crystal, κ and $d\theta/dx$ represent the curvature of the deflection of the beam, which in turn corresponds to the static deflection Fig. 4(b).

The photothermal analysis showed that the temperature profile through the thickness of the quartz has a local skin-effect characteristic, i.e., the temperature drops exponentially and falls aways within the first few tens of micrometers through the thickness, from the nichrome quartz interface as shown in Figs. 3 and 5. Due to this effect, a second analytical method was used which assumes the quartz crystal being artificially divided into two materials, each having different linear expansion coefficient α_1 and α_2, bonded together at a layer (which is few tens of micrometers away from the quartz–nichrome interface, through the quartz thickness, the z-axis), yielding a thickness t_1 at 25 μm and thickness t_2 at 100 μm where $t_1 + t_2$ is the total thickness of the quartz as shown in Fig. 4(b). Since the 100-μm thickness is experiencing no rise in temperature, then the linear thermal expansion coefficient of this thickness is assumed to be zero and the linear thermal expansion coefficient of the 25-μm thickness to be that of the quartz. This is analogous to a bimetallic strip. The curvature of the crystal due to such an effect is given by [16]:

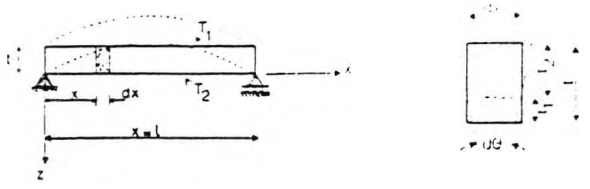


Fig. 4. Schematic of thermal effect on the beam. (a) Temperature gradient applied to the quartz beam. (b) Curvature due to the thermal effect.

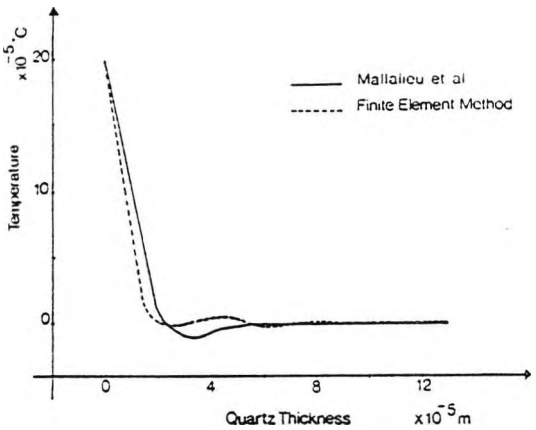


Fig. 5. Temperature profile through the quartz plate as a function of its thickness.

$$\kappa = \frac{6(\alpha_1 - \alpha_2)T}{\dfrac{(E_1 t_1^2 - E_2 t_2^2)^2}{E_1 E_2 t_1 t_2 (t_1 + t_2)} + 4(t_1 + t_2)} \tag{6}$$

where

- T temperature rise of the bimetallic strip,
- κ curvature,
- α_1 linear thermal expansion coefficient of top material,
- α_2 linear thermal expansion coefficient of bottom material,
- E_1, E_2 the Young modulus of both thickness (equal in this case),
- t total thickness of the crystal ($=t_1 + t_2$).

In both analytical methods described above, the radius of the curvature R is given by

$$R = 1/\kappa. \tag{7}$$

From simple geometry, and for given geometrical and physical properties of the quartz crystal, the lateral deflection can be calculated. The dimensions used herein approximate to those of the experimental configuration of Mallalieu *et al.* [9] and McGlade *et al.* [4], being slot length l = 7 mm, total breadth b = 1.6 mm, and thickness t = 0.125 mm.

Finally, a model of this representative quartz crystal developed using the finite element method was employed. The finite element method, in general, being an extension of the matrix displacement method, is particularly powerful for solving a differential equation, together with its boundary conditions over a domain of a complex shape [17]. The process, therefore, is to represent the domain by a large number of "finite" elements of simpler shape (if the domain is a complex one). This finite number of elements have associated with them nodal points. The larger the number of nodes per element, the more sophisticated the element. Assuming an approximate variation of the function of interest over an element, the function can be obtained in terms of the nodal values of that function for a particular element.

In this way, various important parameters were determined, namely the resonant frequency and the change in that frequency due to applied load by introducing an initial stress in the beam, as well as the static deflection by assuming a temperature loading. In choosing an element, it is necessary that the function (displacement being the parameter of interest in this case) converges with an increase in mesh refinement, i.e., with an increase of the outlines of the elements used to model the object (structure) of interest. The result of this converging (confirming) type of element is shown in Fig. 6(a), but a nonconverging type of element will result the behavior shown in Fig. 6(b), while Fig. 6(c) shows a partially converged function. The use of an element of the type producing such behavior should be avoided. Hence, if any type of element is chosen, the discritization of the structure has to be made such that a badly shaped element, (among the chosen type), does not occur, i.e., keep the shape of the chosen type of element as equilateral as possible. Otherwise numerical errors in the calculation can occur if a badly shaped element is chosen, such as nonequilateral triangle, or a distorted square element, especially at its corners.

The quartz crystal was assumed to be a thin plate structure and was discretized into 52 semiloof elements as shown in Fig. 7, having 8 nodes each (one at each corner and four at midsides). The structure was assumed to be clamped at both ends in a similar way to the experimental configuration and temperature loading was implemented by assuming a temperature gradient through the thickness of the quartz plate. The displacement at the center of the structure was monitored. The 52-element mesh was increased by approximately a factor of 4 for better accuracy of the results. The displacement calculated thereby was almost the same as in case of the 52-element mesh.

Since the quartz crystal was considered to be experiencing a flexural vibration in its first mode of vibration, at the resonant frequency, the system under consideration could be approximated to single degree of freedom under forced vibration. The maximum deflection under dynamic conditions was then given by multiplying the static deflection by the Q-factor of the system, where the static deflection is calculated by each of the three methods described above and comparisons made.

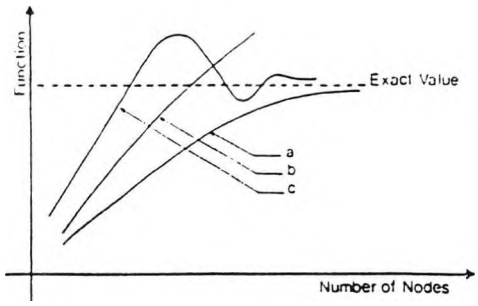


Fig. 6. Convergence of a function using finite element method. (a) Converging function, (b) nonconverging function, (c) a partially converging function.

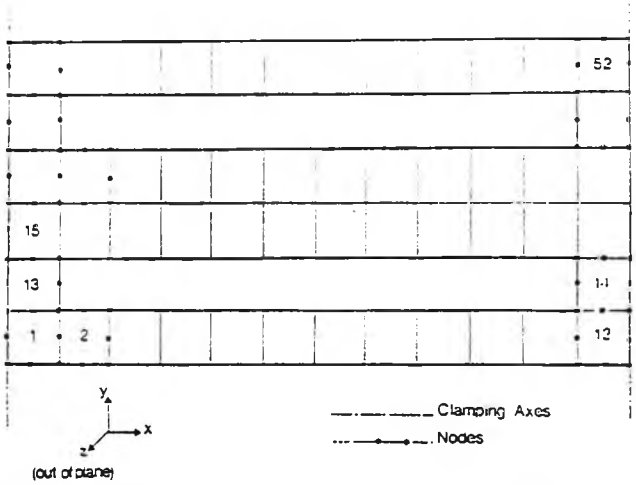


Fig. 7. Mesh of the structure analysis of the quartz plate showing nodes and elements.

III. Results

A. Photothermal Effect: Sensitivity and Duty Cycle

An analysis, similar to that of Mallalieu *et al.* [9] was carried out at a fixed average input power. This analysis was applied to explore the sensitivity to the nichrome thermal conductivity, as the temperature rise at the nichrome–quartz interface is greatly dependent on the thermal conductivity of the material with which the quartz is coated, in the evaporation process during the preparation of the sensor, which can lead to a variable composition of the layer. It was found that a 100-percent increase in thermal conductivity of the nichrome yields to a 45-percent decrease in the temperature rise at the quartz–nichrome interface. This in turn governs the maximum deflection experienced. The temperature profile for the distribution of heat through the quartz thickness was also extracted using the finite element method and this showed a similar distribution to that given by Mallalieu as seen in Fig. 5 where an exponential decay results and the temperature (ac component) drops to zero at a distance of about 25 μm through the quartz thickness. A one dimensional heat flow was assumed in the analysis above. Recalling (3), the magnitude of the fundamental component behaves as a sinc function, i.e., as the duty cycle decreases, the magnitude of the fundamental component also increases. This leads to a possible useful future experimental investigation of the driving of the sensor at the lowest possible practical duty cycle (for a designed deflection) whereby pulsed-laser diodes can be used at low average power, and thereby lower the cost of the source.

B. Resonant Frequency and Deflection Mode Shape

Using the finite element model, the value of the resonant frequency calculated was as expected, in agreement with the analytical values obtained from (1), for the same crystal geometry, physical properties, and boundary conditions. A value of 10.3 kHz for an unloaded crystal is reported. The same system was solved to determine the change in frequency due to the applied load by applying an initial axial stress to the structure.

Fig. 8 illustrates a graph of the resonant frequency predicted as a function of load, together with, for comparison, the experimental measurement of Mallalieu [13] for a similar crystal. The latter experimental condition cannot be modeled exactly as precise dimensions of the quartz crystal are not available. The (t/l^2) ratio especially affects the frequency experienced by the quartz, as seen from (1) and (2). Thus, say a 10-percent increase in the effective length of the quartz crystal due to the experimental uncertainties and a 4-percent decrease in thickness due to manufacturing limitations, will cause the unloaded resonant frequency to decrease by approximately 23 percent. Within this constraint the theoretical result shows close similarities to the experimental. A linear relationship is seen, as well as an approximate similarity in the slope of the theoretical frequency–load relationship to that observed experimentally, being $4.9 \times 10^{-4} f_0\, g^{-1}$ and $5.3 \times 10^{-4} f_0 g^{-1}$. This gives agreement to within 8 percent and it is important as the linear relationship of frequency to load is a particularly useful feature of the quartz resonant sensor. The limited theoretical analysis of Kirman [12] also revealed a discrepancy of the same order between the calculated value of resonant frequency and that observed experimentally

Fig. 9 shows the mode shape of the fundamental component, illustrating the calculated displacement of various points along the crystal, where as expected, the maximum deflection is at the center of the quartz crystal. This result is in agreement with the mode shapes derived theoretically in the simpler analysis of Albert [6] and is at variance with the mode shape suggested by Mallalieu [9].

C. Deflection

In this work, the deflection can be calculated assuming any arbitrary value of optical power. A value of 4-mW average input power as used by Mallalieu [13] was used. Three different methods were considered. Equations (5) and (7) yield the curvature of the deflected quartz crystal calculated assuming a linear temperature gradient through the thickness of the quartz. This method results a static deflection Z_s of 0.24 nm.

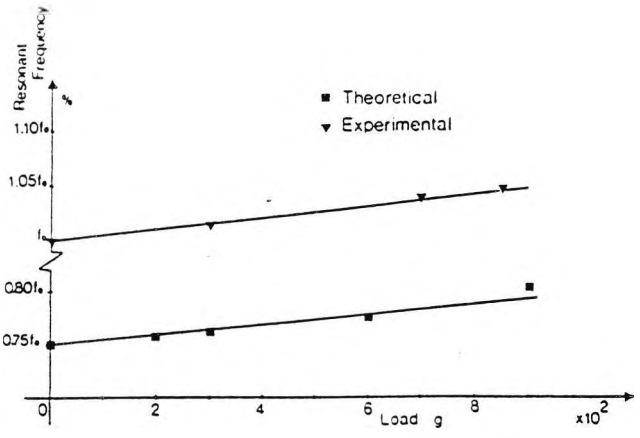


Fig. 8. Graph of resonant frequency as a function load applied (theoretical and experimental).

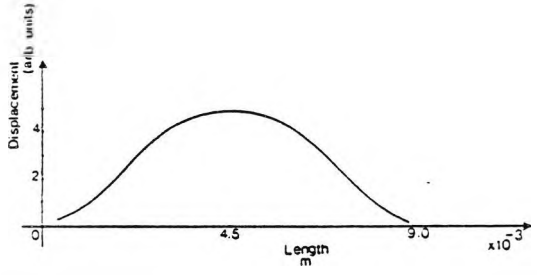


Fig. 9. Predicted first mode shape along the quartz plate using finite element method.

An alternative approach is to implement the bimetallic strip theory and assume the crystal artificially divided into two materials of different linear thermal expansion coefficients. From (6) and (7), a static deflection of 0.22 nm can be calculated which is in close agreement with the previous calculation.

The third and final method used to calculate the static deflection was the finite element method. Applying a temperature load of the same profile through the crystal thickness as that of Mallalieu [13], it was found to give a maximum displacement of 0.10 nm.

Recalling that the quartz crystal is experiencing a flexural vibration in its first mode at the resonant frequency of the structure, and approximating the system to a single degree of freedom under forced vibration, the maximum deflection Z_{max}, (at the center of the quartz crystal), under dynamic conditions is 0.65 and 0.59 μm, respectively, from the first two methods, and 0.27 μm using the finite element method. This assumes a reported value of Q of 2700 for the quartz [4], and

$$Z_{max} = Z_s Q. \tag{8}$$

This latter result is of the same order of magnitude as previous results and indicates the high confidence that can be placed on such a calculation. This is in closer agreement with a value of "a few micrometers" for 5.5-mW average input power which had been reported from experimental measurements [18], in contrast to a theoretical deflection of 50 μm reported by Mallalieu *et al.* [9].

Additionally the use of an optical element ≤ 1 μm from the crystal in the evanescent wave coupled device of Grattan *et al.* [10] implied a submicron movement of the crystal (for an optical power of ≤ 1 mW with ~20-percent optical/electrical conversion), otherwise its movement would be severely damped by collision with the input fiber.

IV. Discussion

Several mathematical models were developed to investigate theoretically the behavior of a quartz resonant structure, similar to that for which experimental results were available, and comparisons made.

The uncertainty of the dimensions of the quartz crystal used in the experimental configuration does lead to a significant difference in calculating the resonant frequency and the frequency-load relationship, due to the dependence of the resonant frequency on the thickness (t) of the quartz crystal and the length squared (l^2). However, both the unloaded resonant frequency and the frequency-load relationship calculated were in good agreement with experimental values, within the dimensional uncertainties of the crystal itself.

These factors mentioned may well affect the theoretical calculations of the maximum deflection which occurs at the center of the quartz. However, the figures obtained for this deflection are of the same order of magnitude as those measured experimentally [18], whereas another theoretical approach reported by Mallalieu *et al.* [9] had shown a difference by a factor of up to 50. This shows the validity of the theoretical approach described herein.

In the study of the photothermal analysis, it was shown that, for a fixed input power and fixed geometrical and physical properties of the sensor, the material used in the evaporation process may be a significant factor governing the maximum deflection. This is due to the possible variations in the composition of the actual material coating the quartz.

Finally, an additional study was carried out to investigate the effect of the duty cycle of the input signal by implementing a Fourier analysis and assuming a constant average input power. This analysis showed that the magnitude of the fundamental component of the input signal (which is linearly related to the displacement) being dependent of the duty cycle implies that the most economical use of the average input power is for the lowest practical duty cycle, to give a specific displacement.

Overall, this work has shown the utility of the techniques described to investigate the resonant sensors discussed. Such an approach has an obvious application in the design of new sensors using the phenomenon described and especially in the consideration of microstructures of comparable dimensions to optical fibers themselves in quartz or other materials [19].

Acknowledgment

The authors are pleased to acknowledge helpful discussions with R. Gay and P. Steel, Marconi Research Cen-

ter, Chelmsford, U.K., and with Dr. M. K. Mirza of City University, London.

References

[1] T. G. Giallorenzi, J. A. Bucaro, A. Dandridge, G. H. Siegel, Jr., J. H. Cole, S. C. Rashleigh, and R. G. Priest, "Optical fiber sensor technology," *IEEE J. Quantum Electron.*, vol. QE-1, no. 8, pp. 626-664, 1982.

[2] K. T. V. Grattan, "Recent advances in fiber optic sensors," *Measurement J. Int. Meas. Confed.*, vol. 5, pp. 112-134, 1987.

[3] G. D. Pitt, P. Extance, R. C. Neat, D. N. Batchelder, R. E. Jones, J. A. Barnett, R. H. Pratt, "Optical fiber sensors," *Proc. Inst. Elec. Eng.*, vol. 132J, pp. 214-247, 1985.

[4] S. M. McGlade, G. R. Jones, "An optically powered vibrating quartz force sensor," *GEC J. Res.*, vol. 2, pp. 135-138, 1984.

[5] Marconi Research Center, "An optically driven and optically read force sensor," Techbrief, GEC Research, 1986.

[6] W. C. Albert, "Vibrating quartz crystal beam accelerometer," Inertial Research Department, The Singer Company—Kearfott Division, Fairfield, NJ, 1982.

[7] N. R. Serra, "Quartz resonator digital accelerometer," Tech. Rep., C.A. Litton Systems Inc. Woodland Hills, CA; also in *AGARD Conf. Proc.* (Oxford, England), 1967, p. 43.

[8] J. M. Paros, "Precision digital pressure transducer," *ISA Trans.*, vol. 12, pp. 173-179, 1973.

[9] K. Mallalieu, R. Youngquist, D. E. N. Davies, and G. R. Jones, "An analysis of the photothermal drive of a quartz force sensor," in *Proc. Fiber Optics '85* (Chistlehurst, Kent, England), 1985, pp. 211-218.

[10] K. T. V. Grattan, A. W. Palmer, D. P. S. Saini, "Optical vibrating quartz crystal pressure sensor using frustrated-total-internal reflection read out technique," *J. Lightwave Technol.*, vol. LT-5, pp. 972-979, 1987.

[11] E. Dieulesaint, D. Roger, H. Rakouth, "Optical excitation of quartz resonators," *Electron. Lett.*, vol. 18, pp. 318-312, 1982.

[12] P. G. Kirman, "A vibrating quartz force sensor," presented at Transducer Tempcon. Conf., London, England, 1983.

[13] K. I. Mallalieu, "Investigations into the optical activiation and fiber optic multiplexed interrogation of vibrating sensors," Ph.D. thesis, University of London, U.K., Apr. 1987, pp. 38, 65.

[14] K. T. V. Grattan, A. W. Palmer, D. P. S. Saini, "Frustrated total internal reflection fibre optic pressure sensor," *J. Lightwave Technol.*, vol. LT-3, pp. 1130-1134, 1985.

[15] J. M. Gere and S. P. Timoshemko, *Mechanics of Materials*, 2nd ed. New York: Wandsworth Inst., 1985.

[16] L. E. Andreva, *Elastic Elements of Instruments*, translation of Upcugie Elementy Priborov (in Russian). Moskva, U.S.S.R.: IPST Ltd., pp. 180-182, 1962.

[17] C. T. F. Ross, *Finite Element Methods in Structural Mechanics*. West Sussex, England: Ellis Harwood Ltd., 1985, pp. 11-15.

[18] R. Gay, Marconi Research Center, Chelmsford, U.K., private communication, 1987.

[19] S. Venkatesh and S. Novak, "Micromechanical resonators in fiber-optic systems," O.S.A. *Opt. Lett.*, vol. 12, pp. 129-131, 1987.

*