UNIVERSITY OF SOUTHAMPTON

Shape Optimization of Turbine Blade Firtrees

Wenbin Song

宋 文 滨

A dissertation submitted in partial fulfillment

of the requirements for degree of

Doctor of Philosophy

COMPUTATIONAL ENGINEERING AND DESIGN CENTRE

SCHOOL OF ENGINEERING SCIENCES

October 2002

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

SCHOOL OF ENGINEERING SCIENCES
COMPUTATIONAL ENGINEERING DESIGN CENTRE

Doctor of Philosophy

SHAPE OPTIMISATION OF TURBINE BLADE FIRTREES

by Wenbin Song

The effective application of various optimisation techniques including classic gradient-based and modern evolutionary computation methods in engineering design practice can not only deliver better quality, but also shorten design cycle time. However, the success of using these techniques relies on a number of factors, such as efficient design parameterisation, complete automation, and expertise in deploying various search tools and managing the high computational cost associated with the use of high-fidelity simulation code. A CAD-based shape optimisation method is investigated in this work using knowledge-based ICAD® system with focus on the optimum shape design of turbine blade firtrees. The design of such a structure component involves a large number of constraints derived from industrial experience. The overall aim of this work is to employ some effective and efficient search techniques to explore various new shapes based on an automated design-to-analysis integration, which is achieved by incorporating a knowledge-based intelligent computer-aided design system (ICAD) into the process using sequential rule-based modelling methods. Analysis-related information as well as geometric data is integrated together to produce a general template for the firtree. A high-fidelity finite element analysis code is used as the assessment tool of structural strength and different types of stress criteria are used in the formulation of the optimisation problem. Both the existing shape features inherent to CAD systems and new features offered by the use of free-from shape modelling using Non-Uniform Rational B-Splines (NURBS) are investigated. This leads to a combined feature-based and free-form shape parameterisation method. A two-stage (Genetic Algorithms + Local Search) procedure is used in order to make use of the advantages offered by these two methods while overcoming some of their weaknesses. The problem of high computational cost problem is also tackled by the efficient use of a Gaussian Process based surrogate model coupled with Genetic Algorithms. Both the combined shape parameterisation methods and framework for incorporating surrogates with GA can be applied to general engineering design problems.

Acknowledgements

I would like to thank my supervisor Professor Andy Keane for his constant support and useful discussions during the study, especially in the first year. I would also like to thank Dr. Atul Bhaskar for his timely guidance and useful suggestions.

A special thank goes to Miss Janet Rees for her assistance, not only technical, but also the effort for arranging many visits to Rolls-Royce and also other engineers and specialists from the company for their concrete assistance and numerous communications I made through either email or personal visits to the company in the past three years during the research activity.

I am grateful to my colleagues at the CEDC. In particular, I would like to thank Prasanth Nair, Yew Soon Ong, Yu-Ming Chen, Mamahdou Bah, Yong Zhang and many others for their support.

I would gratefully acknowledge the financial support received from both the EPSRC under contract GR/M53158 and from the University Technology Partnership for Design, which is a collaboration between Rolls-Royce plc. BAE Systems, and University of Cambridge, Sheffield, and Southampton.

And also special thanks to my parents and my wife Yan Qiao, for their continuous encouragement and support during the last three years.

Wenbin Song

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Overview of design, analysis and optimisation

Over the last several decades, the engineering design world has been transformed by the introduction of massive computational power. Computers are now a principal tool throughout the whole product lifetime including conceptual design, analysis, mock-up, manufacture, operation, and maintenance, etc. Design, analysis and mock-up are now completely carried out in a virtual environment and this process is being continuously updated not only in the complexity of the models within a particular domain but also in a multi-disciplinary sense. In the engineering design community, the most notable changes lie in two aspects: computer aided design (CAD) and computational analysis. Most companies use CAD and analysis software to deliver more reliable products in increasingly reduced time scales, but the pursuit of lower cost, better design and shorter development time never stops because of the competitive world market.[1] It is believed that design decisions made at earlier stages have much greater effects than those made at later detailed design stages, therefore it is desirable to include as much detail as possible into the earlier models.

The design process, in general, is a recursive one in which changes are often required during the later stages due to various factors arising as the design progresses, and design requirements often not only involve structure and functionality, but also cost, manufacture and environmental aspects. The increasing complexity of engineering systems, not only in single disciplines, but also in the interactions between different disciplines tends to make this process even longer. The solutions for these increasing complexities and interactions could not be achieved without the effective and efficient use of today's various rapidly updated computing techniques, noticeably, design optimisation methods. In addition, the use of artificial intelligence methods in engineering design has attracted a lot focus in recent years. Various techniques have been applied in the field of product design from earlier expert systems to most recently knowledge-based systems. Many traditionally designed analysis system such as ANSYS[2] are also introducing these new techniques to enhance the capability to offer integrated design tools, for example, the Design Space module introduced in ANSYS version 6.

The use of CAD tools and computational simulation techniques such as finite element analysis and computational fluid dynamics has become common practice in the engineering design community. It is also known that utilization of various numerical optimisation techniques can bring benefits in terms of improvement in product quality and reduction in design cycle times. The use of detailed high-fidelity analysis in optimisation will bring more confidence in the design. In general, there are two scenarios in which optimisation techniques can be used: 1) during the preliminary design phase, where many design configurations need to be considered to identify a design which meets the requirements as well as to accommodate innovation, 2) once the configuration has been determined, at which point the detailed dimensions of the component can be optimised against a set of detailed constraints like stresses, and cost. Optimisation techniques have typically been used in structural optimisation where weight reduction is chosen as the objective, subject to a number of stress constraints at points within the structure.

The optimisation process typically starts with the parameterisation of the model and is then followed by a search process using different algorithms and strategies based on the evaluation of a measure of merit. A typical structural optimisation problem commonly involves sizing, topology and shape optimisation. Sizing optimisation is used to find the size related variables, such as cross-sectional area for bars and trusses. Topology optimisation mainly involves the determination of optimum configuration usually starting with a block of material and shape optimisation considers the optimum shape of the component boundaries. Various parameterisation approaches have been reported in the literature. Sarameh[3] provided a comprehensive survey on the shape parameterisation approaches and related sensitivity analysis and mesh generation, deformation issues in the multidisciplinary domain. Some preliminary guidelines regarding the choice of parameterisation methods were also provided in the paper.

Eight different types of parameterisation methods were reported by Sarameh, including basis vector, domain element, partial differential equation (PDE), discrete, polynomial and spline, CAD-based, analytical, and free-form deformation (FFD). These methods can be generally classified into three different types. The first is based on using the coordinates of the boundary nodes in a discretized domain as design variables, which is relatively easy to implement but difficult to maintain a smooth geometry and the resulting optimised designs may be impractical for manufacture. The second is based on using a CAD system. Primitives, such as circular arcs, rectangles, provided in CAD systems are used as building blocks in geometry modelling. These primitives are also known as features. Parameters describing these features are usually chosen as design variables in optimisation. Although the advantages of using a CAD system is obvious: consistent with engineering practice, relatively small number of design variables compared to node-based shape parameterisation, it is difficult for some CAD systems to handle the variations of topology due to large perturbations in some dimensions. In addition, the underlying parameters describing the geometry are not normally available to calculate the sensitivities analytically. The third approach is similar to the morphing techniques used in computer sciences, and is termed free-form deformation (FFD),[4] in which, the deformation instead of the base geometry is modelled. In general, shape optimisation problems can be further classified into two types. The first type involves the determination of dimensions of pre-defined shapes (geometric features) such as the radius of circular hole, etc. This type of design method can be referred to as feature-based design. The second type involves the determination of the shape of an arbitrary open or closed boundary. In this case, shape can be represented as a combination of prescribed shapes via the use of a basis vector to reduce the number of design variables used in defining the shape.[5] In the domain of computational fluid dynamics (CFD), perturbations instead of the original shapes can also be used in modelling the geometry because of the large effect of small variations on the aerodynamic performance, as reported by Lépine *et al.*[6]

Among various parameterisation methods, the CAD-based approach provides a natural way for geometry modelling, especially for multidisciplinary applications, where dissimilar grids often used in different domains can be extracted from the same CAD geometry. It is, however, very difficult, for many CAD systems to capture design intents and to describe the functional relationships between thousands of entities in a complex product model. What is produced by this kind of CAD system is the final results of design, not the modelling process, which is often documented separately, if at all. Whenever the designer would like to review the design or rebuild the modelling process with even minor modifications it usually takes a long time. This kind of knowledge is rarely documented or most often is identified as part of the experience of designers and is difficult to utilize during subsequent iterations by other designers. Modern CAD systems can provide template-like parametric models, which will produce different geometries based on different input parameters. However, a parametric model itself does not solve all the problems identified above. Knowledge-based engineering and expert systems have made some progress in this regard. The basic idea behind "knowledge-based engineering (KBE)" is to find ways to record different kinds of knowledge about how to design, configure and analyse a product. A primarily declarative KBE language is usually used to capture knowledge in the form of "rules". The programmed rules specify how various parameters and attributes of objects in a model are related to each other. These rules can be expressed in many forms including math and logical formulas. The intelligent CAD system (ICAD®) from Knowledge Technologies International (KTI)[7] is a combination of knowledge-based engineering and CAD technology and its generative template modelling capability can be used to provide flexible and robust geometries for subsequent analysis.

1.1.1. Design automation using knowledge-based engineering

The difference between using conventional CAD and using a KBE-based approach lies in the following aspects: KBE is knowledge oriented, while CAD is geometry oriented; CAD works from the bottom up with detailed dimensions while KBE works on conceptual level and can harness all the design specifications including physical laws, material criteria, manufacturing constraints, and even non-technical aspects. KBE uses rule-based model generation methods in conjunction with a CAD modelling engine supporting solid modelling techniques. The rules used to describe the geometric relationships can be expressed in various forms, such as equalities or inequalities using mathematical and logical relationships. It can further make some decisions about the design on the basis of the rules supplied to it. This gives the model some kind of intelligence. Features offered by the object-oriented methods make it possible to provide rapid modification, backward compatibility, etc. By using ICAD, the savings in time, cost and man-hours can be dramatic. Another benefit is the ability to maximize quality, accelerate innovation, and optimise performance.

In this work, CAD-based approach is adopted but the parametric modelling is performed using ICAD from KTI.[8] Although geometry is the main object of manipulation, the most interesting feature is that ICAD can be used to capture various knowledge such as the best practice, performance data, manufacturing process and cost criteria into a complete product model, known as a "generative model". For example, non-geometric information such as loading conditions, material properties, etc. can be integrated into the model and used in the following analysis stage. Geometric entities in an ICAD model are programmable objects, which, once completely defined, can be repeatedly used in higher-level models. It is this feature that enables ICAD to provide greater flexibility than other CAD systems. The geometric modelling engine and geometry exchange standards supported by ICAD makes it possible for smooth transferring of geometry to analysis code. The subsequent incorporation of ICAD and finite element codes into a search procedure gives us the ability to carry out search based on high fidelity analysis results and to explore different geometric features using a relatively small set of parameters.

Here, the basic procedure for the design of the firtree geometry using ICAD falls into two steps: first to identify the features and rules used to define the geometry and second to break down the whole model into several modules each of which becomes a building block in the hierarchical structure of the model. In ICAD, each of these basic blocks is described using the ICAD design language (IDL) as a generic definition, which can be implemented in the ICAD browser using a specific set of parameter values. Thus the model is defined parametrically: different sets of parameter values will result in different designs from the same template. In addition, multi-modality and backward compatibility can be achieved by incorporating different behaviour into one model and maintain the interface unchanged while the internal implementation is modified due to various reasons such as software upgrading or improved algorithms.

1.1.2. **Optimisation techniques and strategies**

Many techniques in design optimisation originated in the field of structural optimisation. Structural optimisation has been a topic of interest for over 100 years. A comprehensive overview was provided by Vanderplaats[9] on the use of optimisation techniques for structural design. A systematic treatment on the concept and applications of optimal design can be found in Ref. 10. The study and development of computer programs on mathematical optimisation algorithms have been a long-term effort since the early 1960's and a large number of numerical algorithms are now available both in the form of standard function library APIs and some design exploration systems. An integrated general data management and optimisation package called OPTIONS[11] developed by Keane is still being regularly updated. A concise description on the available methods in the OPTIONS can be found in its manual[*]. OPTIONS is a system which provides a flexible framework for incorporating user codes ranging from very simple ones to complicated external software packages as well as more than 40 search algorithms and can be used either interactively or in batch mode. Commercial packages such as iSIGHT[12] also exist which in general provide good Graphical User Interface (GUI) and technical support but with a relatively small number of optimisation methods implemented. And some methods have also been implemented in commercial finite element tools such ANSYS. However, the use of numerical optimisation techniques in the current engineering design practice is by no means comparable to the maturity of various search techniques developed over the past several decades and optimisation capabilities are often strongly coupled with the analysis package, which itself is quite a challenge for designers to use. Recent developments and trends in global optimisation can be found in Pardalos *et al.*[13]

Different types of search techniques are available, these include classic gradient-based search methods such as Newton, quasi-Newton method, Sequential Quadratic Programming (SQP), etc. and gradient-free methods which include the class of direct

---

[*] http://www.soton.ac.uk/~ajk/options.ps

search, in which, pattern search has attracted large amount of attention. The other important class of gradient-free methods are the evolutionary computation methods. A good review on various search techniques can be found in Neumaier.[14] Lewis *et al.* provided an overview on direct search methods.[15] These methods are referred to as classic optimisation methods. Different types of modern stochastic methods have been developed and applied in wide range of domains over recent years. These include Genetic Algorithms (GAs),[16] Simulated Annealing (SA),[17] Evolutionary Programming (EP),[18] and Evolutionary Strategy (ES).[19] An online methods guide and available software codes covering most classic linear and non-linear algorithms can be found on the NEOS server for optimisation.[20]

The automated design-to-analysis is then further integrated into an optimisation procedure to try to improve the design. The use of high-fidelity analysis code in optimisation is always desirable as it can provide a more accurate estimate to the quantities that decide the quality of the design. However, the high computational cost associated with the use of high-fidelity code motivated the research on the use of approximation or cheap models in optimisation. This lead to different approaches based on whether the surrogate models are used or not in the procedure. Conventional methods make direct calls to the expensive simulation code repeatedly. Although these approaches are always more desirable, they can only be used for less expensive simulation codes. Using surrogate or cheap models is sometimes the only choice when the computational cost is high, especially if evolutionary computation methods are also involved. The first step of a typical procedure for this approach is to construct a surrogate model, often a polynomial response surface model, using dataset collected on a number of points in the design space.[21] The choice of design points is often carried out using design of experiment methods (DoE), which basically define these points according to various criteria.[21] One example of DoE methods is the commonly used Latin hypercube sampling. The surrogate models can then be used in a number of different ways, this leads to different strategies. In this work, a strategy coupling a Genetic Algorithm with surrogate models will be investigated.

1.2. Overview of the firtree design problem

The firtree joint is a component often seen in turbine structures to attach a blade to the rotating disc. A typical turbine blade/disc configuration is shown in Figure 1.1. An example of turbine blade is given in Figure 1.2. High mechanical loads generated by the blade are transferred through the joint to the turbine disc. This centrifugal load of a blade is very large and imposes severe requirements on the design of firtrees. Due to size, weight, and stress constraints, the geometric design of a firtree must be fully analysed to obtain the optimum configuration. A systematic treatment on various aspects of the design of such a component can be found in Ref. 22.

The design of such a structure involves a large number of geometric parameters, which have complex relations among them. It is believed that the choice of geometry definition could have great impact on the final design. The most commonly used geometries in the design of firtrees are straight lines and circular arcs.[22] Previous effort was mainly focused on the analysis and experiments of the contact problem involved.[23,24] However, little work has been reported on the shape optimisation of such critical components in the literature. And there is currently a strong need in the industry to automate the design-to-analysis process and bring in optimisation techniques in design. In addition, although different tooth notch profiles have been tried before, for example, an elliptical fillet was tested by Lee *et al.*[25] in order to reduce the peak stresses in notch region, using different sizes for each tooth has not been reported in the literature. This is partly because all the teeth are defined using a single set of parameters, and therefore are not independent.[25] The use of different profile for each teeth could provide potential benefit in terms of more even stress distribution across the teeth because of non-uniform distribution of load on the teeth, which has been known from both finite element analysis and photoelastic experiments.[23,24] In order to achieve a sufficiently flexible geometry definition for optimisation purpose, a modular parametric geometry representation is critical. This model must be able to represent as many as possible different geometries, and thus provide a large design space for optimisers to search.

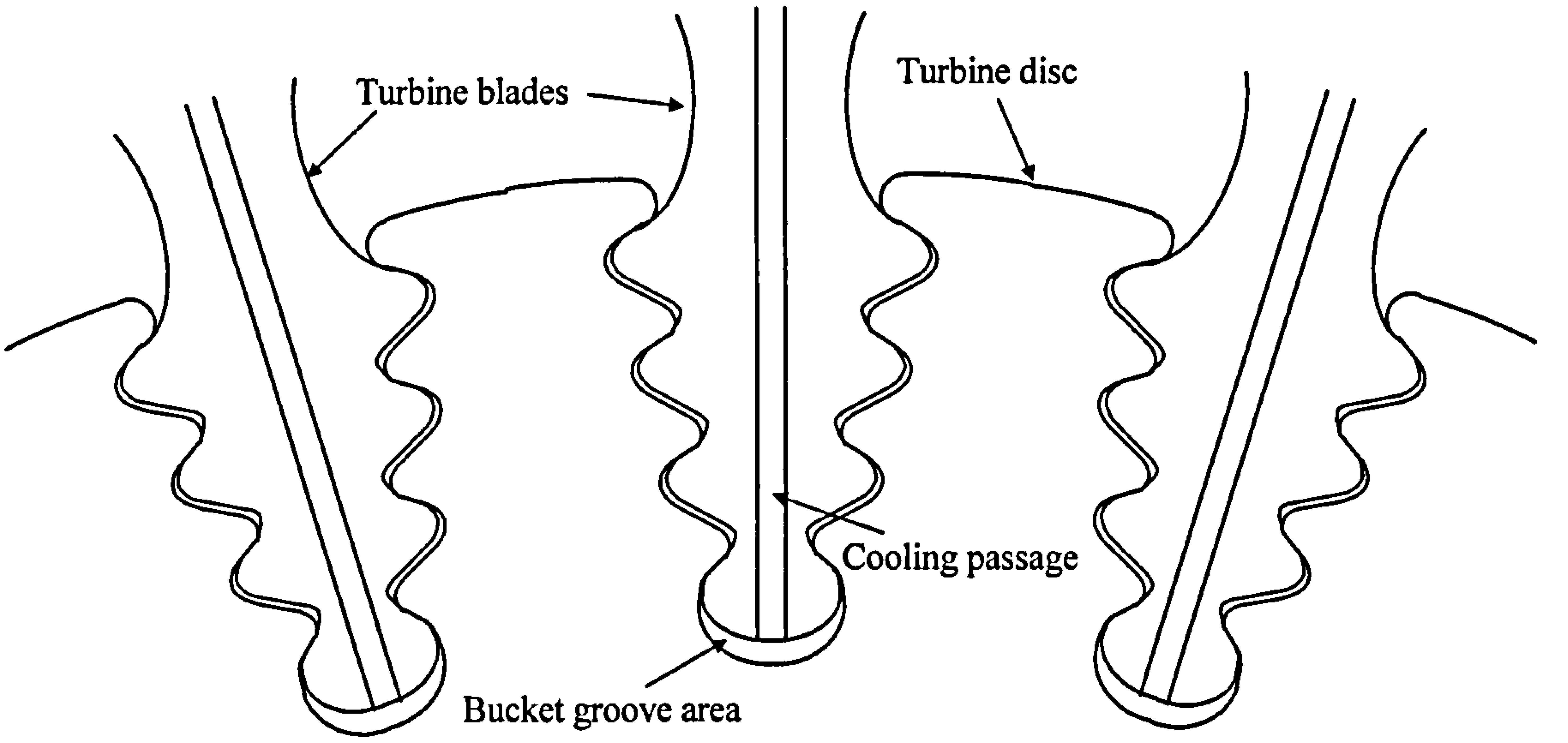


Figure 1.1: Illustration of a typical turbine blade/disc configuration

A number of stress constraints need to be considered in the design of firtrees. These include section stresses on both the firtree root and disc head; crushing stresses which describe the direct tensile stress on the teeth; and peak stresses which occur in the inner fillet radii of both the blade and disc, etc. The average section stress can be estimated assuming an even distribution of total load on each tooth. However, the peak stresses in notch region are difficult to predict without a finite element analysis. A full 3D analysis of the firtree will reveal the stress contours, especially the effect of skew angle on the peak notch stresses. However, the complexity of the model construction and computational burden associated with a full 3D analysis make it unrealistic to include a 3D analysis before any attempt has been made on 2D models. In addition, well-established 2D criteria modified from 3D analysis will probably be adequate.

A systematic parameter study and optimisation cannot be carried out without a fully automated design-analysis capability. The manual approach of attempting different firtree geometries is time-consuming and labour-intensive. It involves several steps such as shape modelling, geometry transfer, stress analysis, and result retrieval, etc. The high cost and slow progress of the manual process have prevented exploration of more complex profiles and different size and shapes for the teeth.

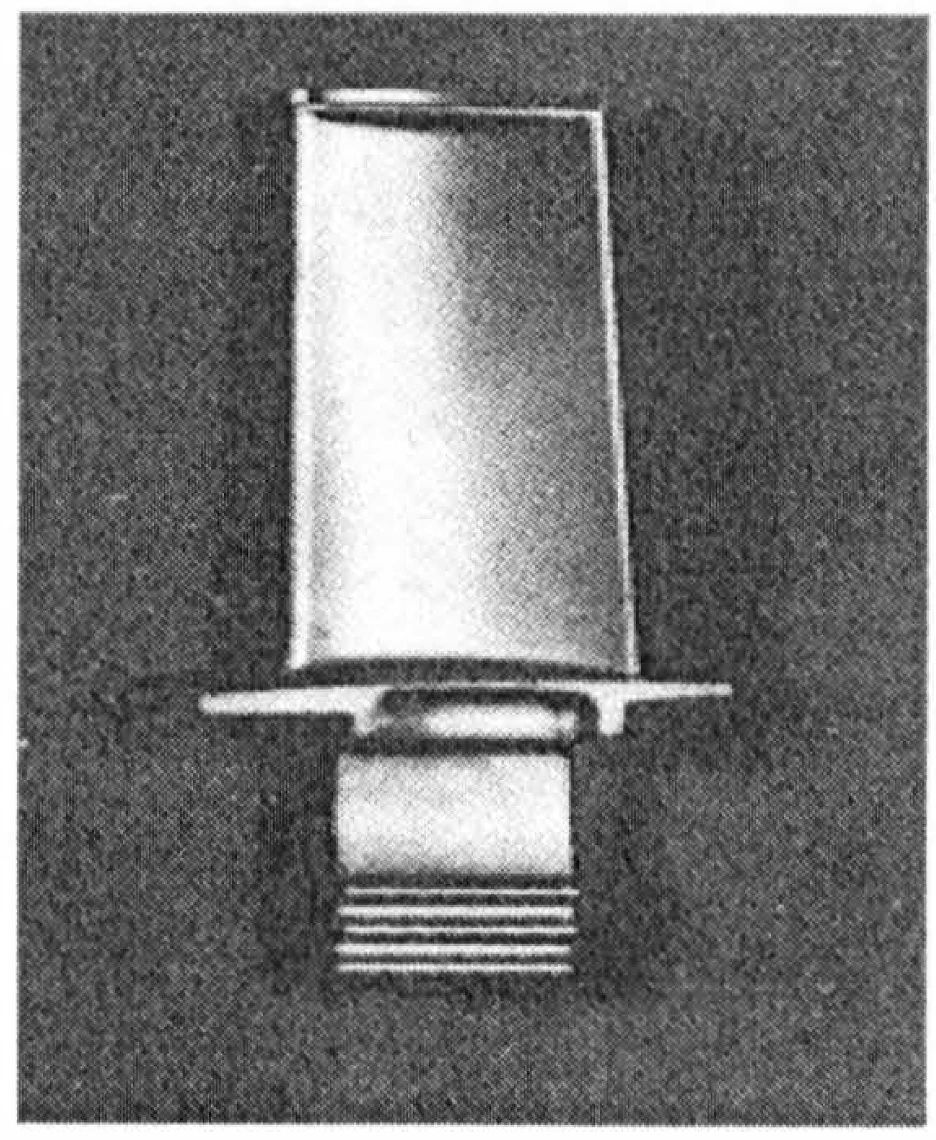


Figure 1.2: A typical Turbine Blade[22]

Therefore, it is critical to establish a direct link between geometry modelling and analysis code. This will enable the designer to explore different candidate geometries at the preliminary stage, ranging from relatively simple designs to rather complex ones, at a reduced timescale. Here, this process involves the use of feature-based parametric geometry construction tools such as ICAD and large-scale structure finite analysis packages, along with an optimisation program implementing various search strategies. Although this problem is a structural optimisation problem, the strategy employed here is rather different. The use of the ICAD system and finite element analysis software together gives it the capability to model the variations parametrically both in the dimension and in the topology and to analyse the effect of geometric features on the stress distribution based on the finite element analysis results. Furthermore, the above process may be incorporated into a search loop to automatically find the best solution against pre-defined goals and constraints. As considered here, this is a 2D problem nested in the overall 3D blade/disc optimisation procedure and uses well-established 2D criteria modified from 3D analysis.

1.3. Research scope and objectives

In order to make use of the vast range of numerical optimisation techniques available today in engineering design practice, parameterisation is the essential first step for optimisation. Among various parameterisation techniques, CAD-based methods proved to be able to provide consistent geometry definitions across the different domains. However, it is difficult for traditional CAD systems to capture knowledge other than geometry itself. A knowledge-based system with geometry modelling capabilities is adopted in this work. Rule-based sequential geometry modelling method is proposed and applied in the shape modelling of turbine blade firtrees. Various features of this method also ease the integration of design, analysis and optimisers, which itself is a challenge in a multidisciplinary design optimisation environment.

Different search algorithms exist for optimisation. Gradient-based search methods provide an efficient procedure for non-linear optimisation problems, but it is usually difficult for these methods to escape the local optima present in many engineering problems. A multi-start method from a set of randomly chosen points is usually adopted to tackle this problem. On the other hand, evolutionary computation methods, particularly Genetic Algorithms, provide a robust and reliable method for finding the global optima at a cost of more function evaluations. In this work, a two-stage search procedure is applied to the formulated complex optimisation problem, in which, a Genetic Algorithm (GA) is first used to identify near optima solutions followed by a gradient-based search to locate the optimum solution at a higher accuracy.

There is a limitation in shape optimisation using existing geometric features provided by the CAD system. It is not possible to come up with new features. Therefore free-form shape modelling using Non-Uniform Rational B-Splines (NURBS) is explored in this work. NURBS promises to be the standard for the geometry modelling community. The introduction of NURBS in shape optimisation essentially overcomes the limitations imposed by feature-based modelling methods, in which the choice of topology and/or features eventually decides the final optimum shape. The difficulties associated with the high number of design variables (coordinates of the control points) when NURBS are used are overcome via the use of non-dimensional coordinates. The combined use of features and local free-form shapes proves to be a general approach and can be used for other shape optimisation problems.

The computational cost associated with evolutionary search procedure is usually high as it typically requires thousands of function evaluations and a typical function evaluation is computationally expensive when it involves high-fidelity analysis models such as finite element models or CFD models. A framework for incorporating surrogate models in genetic algorithms is proposed here and applied to local free-form shape optimisation problem with some success.

1.4. Layout of the thesis

The thesis is organised as follows:

Chapter 2 presents the rule-based geometry modelling method and its application to the firtree shape design following a survey on shape parameter methods in the multidisciplinary domain.

Chapter 3 details the various aspects of the finite element modelling of the firtree root structures.

Chapter 4 describes the optimisation of the traditional firtree shape using a direct link between modelling and analsis codes. A simple yet effective two-stage optimisation procedure is adopted. A Genetic Algorithm is first applied followed by a local gradient search on the promising solutions found so far.

Chapter 5 presents free-form shape representations using NURBS with applications to the local shape modelling in the notch region in an attempt to further reduce the peak stresses. Results using NURBS for notch fillet optimisation are presented.

Chapter 6 describes a GA-based framework using surrogates to improve the efficiency of the original search procedure used in chapter 4. Results on two test functions are presented to illustrate the effectiveness of the proposed framework.

Chapter 7 summarizes the major contributions of the research work and main conclusions; areas for future work are also identified in this chapter.

Chapter 2

Rule-based geometry modelling of turbine blade firtrees

Computer Aided Design (CAD) tools have been widely used in the engineering community. Often it is common that one or more major CAD tools are in use in any engineering company. The capability of CAD tools has evolved from simple 2-D sketching to feature-based parametric 3D solid modelling over the last several decades.[26] It is worth mentioning that it is this parametric modelling capability that makes it possible to incorporate a feature-based parametric modelling system into optimisation process to provide accurate geometry definitions. Moreover, it becomes almost imperative to include a CAD system in emerging multi-disciplinary environments, as more complex geometries will be required when moving from simple test problems to real-world applications. One example of these would be the Framework for Interdisciplinary Design Optimisation (FIDO) developed by NASA.[27] Some challenges and solutions regarding integration of a CAD system into a MDO framework were discussed by Townsend, *et al.*,[28] which mainly include deformation modelling associated with aeroelastic problems, ease of replacement of CAD packages, and sensitivity computations.

A NURBS (Non-Uniform Rational B-Splines)[29] database was introduced in FIDO to tackle deformation modelling by first obtaining the NURBS representation of the CAD geometry, and then deforming the NURBS representation instead of the original CAD model. Townsend *et al.*[28] also suggested that the use of a NURBS database would ease the difficulties associated with the replacement of CAD systems. However, this is not the case as the original geometry is still modelled by the CAD system, and there is no easy way to transform the NURBS representation back to a particular geometry feature as most CAD systems such as Pro/Engineer[30] make extensive use of features. Therefore the choice of CAD system have enormous impact on every aspect of product design and design decisions cannot be made from pure technical point of view. However, it may be argued that a good balance between capabilities and ease of use should be maintained.

An example of direct use of NURBS in coupling between geometry modelling and finite element analysis can also be found in Ref. 31. A geometry-based parameterisation method using the general geometry modeller PATRAN for shape design of elastic solids was presented by Chang, *et al.*[32] Geometry modelling capabilities within the analysis packages such as MSC/NASTRAN and ANSYS can also be used to define the geometries in optimisation problems;[33] and this is current practice for many structural component optimisation problems. In this case, the computation of sensitivities may be easier than using a CAD system as most current CAD packages do not provide sensitivities with shape parameters. But there are also a number of drawbacks in this approach. First, because it is more targeted to analysis capabilities, geometry modelling may be not as good as CAD systems. Second, the transfer of geometry between analysis codes in different disciplines may present quite a challenge. Third some post-processing will normally be required to export the final optimum shape for the purpose of manufacturing. Finally, the search algorithms available in analysis codes such as ANSYS are usually limited compared to the large number of search methods usually provided by general-purpose optimisation packages.

The problem with the adoption of a general-purpose optimisation package in the day-to-day design activity is that designers usually lack the knowledge and experience to use. Therefore, it is usually desirable to provide an automatic link between CAD, analysis and optimisation tools. The first step is to provide a parametric model in optimisation study. The optimisation routines then call upon this parametric model to use it in the traditional form of optimisation problem. Here, an overview of shape parameterisation methods is first given, followed by a description of the rule-based sequential geometry modelling method adopted. Last, the geometry modelling of firtrees is described.

2.1. Overview of shape parameterisation methods for optimisation

Various shape parameterisation approaches exist in the domain of multidisciplinary shape optimisation. Samareh provided a comprehensive survey on these techniques for high-fidelity multidisciplinary shape optimisation.[3] Shape optimisation of open boundaries has been studied for some time using various shape representation methods. For example, in a study carried out by Braibant and Fleury,[34] three shape optimisation problems were examined. These were the determination of the optimal shape of a beam in bending, the optimal shape of a hole and the optimal shape of a fillet and were solved using the concept of design elements which were defined using B-splines composed of a number of elements. In another work, the shape was parameterised using a parametric cubic representation of primitives supported in PATRAN.[35] Equidistant mapping in parametric space of the curve was used to create the finite element discretization. This is also referred to as an iso-parametric mapping in which the curve or surface is discretized evenly in the parametric space. Using these methods, attention must be paid to the creation of internal nodes during FE analysis to avoid the generation of distorted elements. A geometry based approach for coupling CAD with finite element methods was presented where NURBS were used to model the shape of a cross section in a torsion problem.[31] Some more recent work is reported by Waldman, *et al.*[36] and by Schramm *et al.*,[37] in which an optimum free-form shape for a shoulder fillet and beam cross section were obtained. The results in Ref. 36 were presented in tabular form rather than in a way that could be easily used in a CAD-based environment. The use of NURBS can also be seen in CFD-based shape optimisation, for example, the work carried out by Lépine, *et al.*,[38] where it is mainly used to reduce the number of design variables. One of the major drawbacks in the above applications is that the optimised shape is not CAD-ready and additional effort is required to utilize the results in the design process.

The development of parametric feature-based solid modelling capabilities in CAD systems has enabled the implementation of more complex shape optimisation approaches to be developed which output the optimised geometry in a CAD-ready form. Such capabilities have also enabled designers to focus on design innovation by allowing them to quickly develop design concepts utilising the available geometric features. The limitations of this approach are that it is difficult to generate shapes which do not exist in the library of pre-defined features provided by the system. Although free form geometric modelling can alleviate this problem to some extent, it is still very difficult to design a product by purely using free-form geometry, due to the prohibitively large number of control points required. The effect of movement of control points then becomes very difficult to determine. This is particularly important when the design is embedded in an optimisation loop. In this work, the second CAD-based approach is adopted but the parametric modelling is performed using the design automation tool ICAD from KTI.[7]

In this work, a rule-based sequential approach using feature-based design and free-form geometry is adopted in the parameterization of turbine blade firtrees. A typical geometry of blade firtree root and disc head is shown in Figure 2.1 and 2.2, respectively. A firtree shape consisting of straight-lines and circular arcs was first defined and optimised against a number of geometric and mechanical constraints to minimise the firtree weight. The notch stresses have been identified in this study as being unable to satisfy the mechanical constraints but models using single and double arcs in notch provide little flexibility to further optimise the design.[39] Non-uniform rational B-splines (NURBS) are therefore introduced here to define the two-dimensional local tooth profile of the root. A general approach for defining the position of control points is described using non-dimensional quantities that are dependent on the geometric features. One of the advantages of this approach is that flexibility can be achieved without loss of the geometric significance of the design variables used to formulate the optimisation problem. It also provides good control over the range of design variables and the resultant geometry.

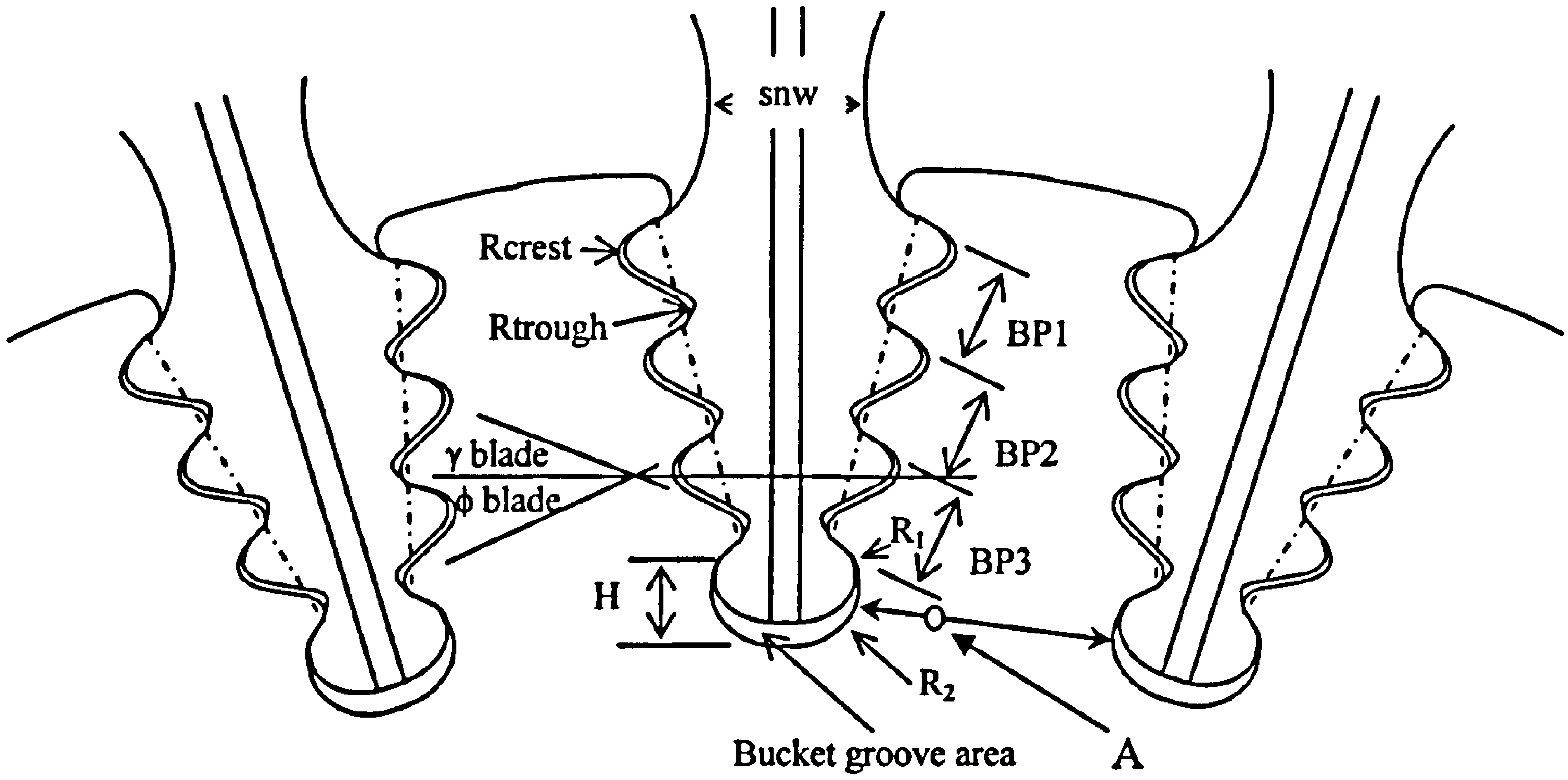


Figure 2.1: Geometry definition of blade firtree root

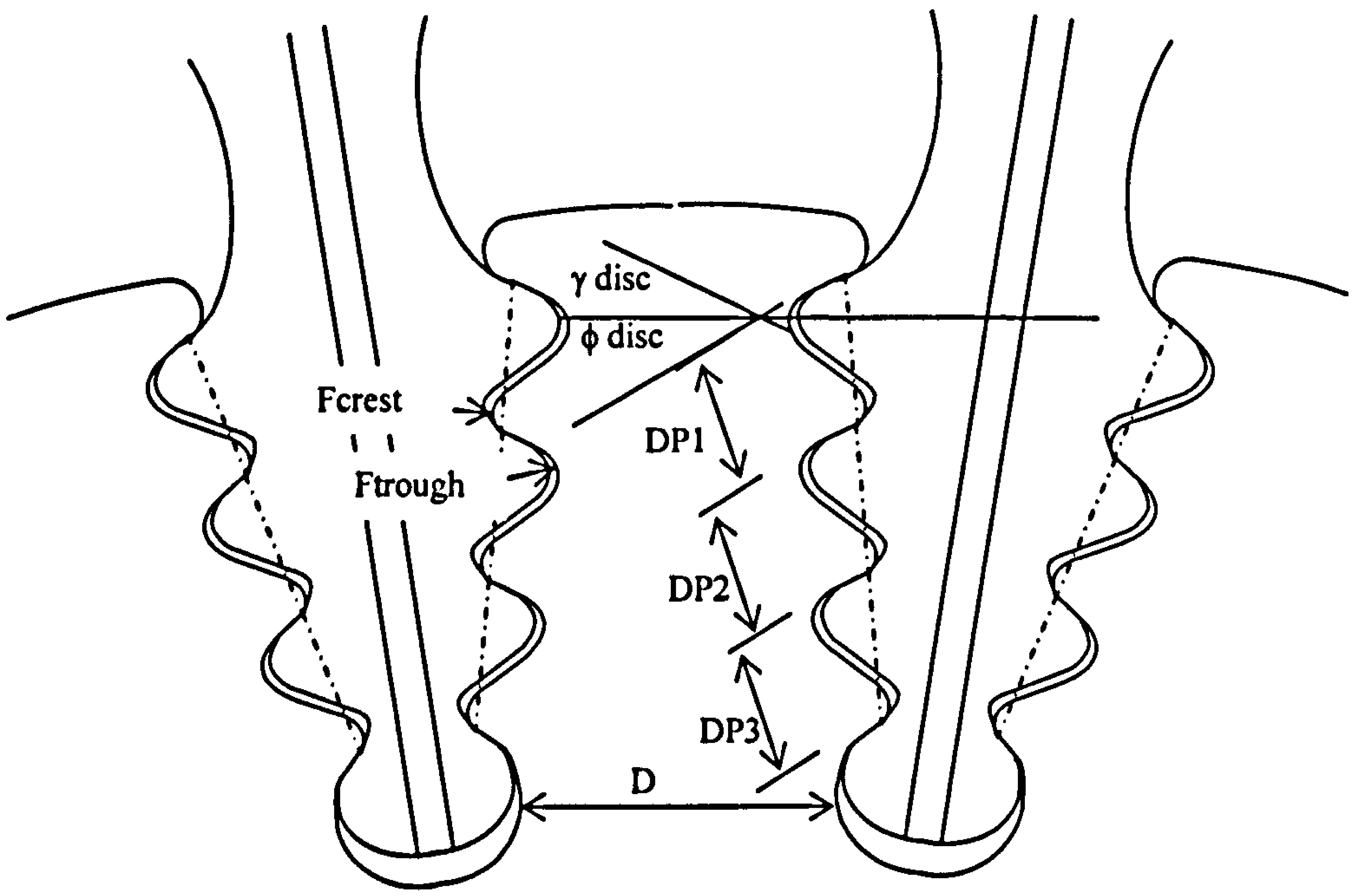


Figure 2.2: Geometry definition of disc head

2.2. Rule-based sequential geometry modelling

At the outset of this work, relations between geometric entities were identified based on both geometric knowledge and design experience. Then a set of independent parameters was chosen as design variables. Design variables are used to define the dimensions of existing geometric entities such as circles. These can be primaries provided by the system, or user-developed components (called Defparts in ICAD). Another type of knowledge is the topology relationships between entities, such as tangent relations between lines and circles. These topological relationships essentially form the underlying rules for the geometry under study.

2.2.1. Sequential geometry modelling of a simple geometry

The position of each entity (Defpart in the ICAD model) must be fully defined before it can be referenced in the definition of other entities. The modelling process is therefore essentially a sequential one, which means that the solving of complex geometric constraints (usually defined in the form of a set of equations) is not automatically supported by the system. A sequential geometry modelling approach is therefore developed here to tackle this problem. The approach utilizes the inter-dependencies between geometric entities to determine a workable modelling process and a suitable range for the parameters. The approach is first introduced on an illustrative example, and then applied to the firtree problem. The details on how the firtree geometry is defined are described next.

The sequential geometry modelling approach may be illustrated by a simple example,[40] as shown in Figure 2.3. This quadrilateral requires five pieces of information to describe its shape. One possible definition of choice would be by four parameters (the lengths of the four edges a,b,c,d) and one topological relation ($AB // CD$). A properly chosen reference is always the first step for geometry modelling. It is always convenient, and without loss of generality, to choose the most straightforward reference point and axis. Then any complex geometric constraints need to be decoupled by the user and transformed into a sequential process. This process essentially involves transforming implicit rules defined by a set of equations into explicit ones (modelling process). The idea of constructive entities is used to assist the process. This can be better explained using this example.

Although the shape in Figure 2.3 can be uniquely determined using the four given input dimensions and one topological relation, the coordinates of the four vertices can not be determined without the solution of a set of geometric constraint equations as listed below: (here without loss of generality, suppose point A is the taken as original of the reference axis and AB is parallel to the horizontal axis, so that, $x_b = a$, $x_a = y_a = y_b = 0$). We call these equations implicit rules defining the geometry.

$$x_d^2 + y_d^2 = d^2 \tag{2.2.1.1}$$

$$(x_c - x_b)^2 + y_c^2 = b^2 \tag{2.2.1.2}$$

$$(x_c - x_d)^2 = c^2 \tag{2.2.1.3}$$

$$y_d = y_c \tag{2.2.1.4}$$

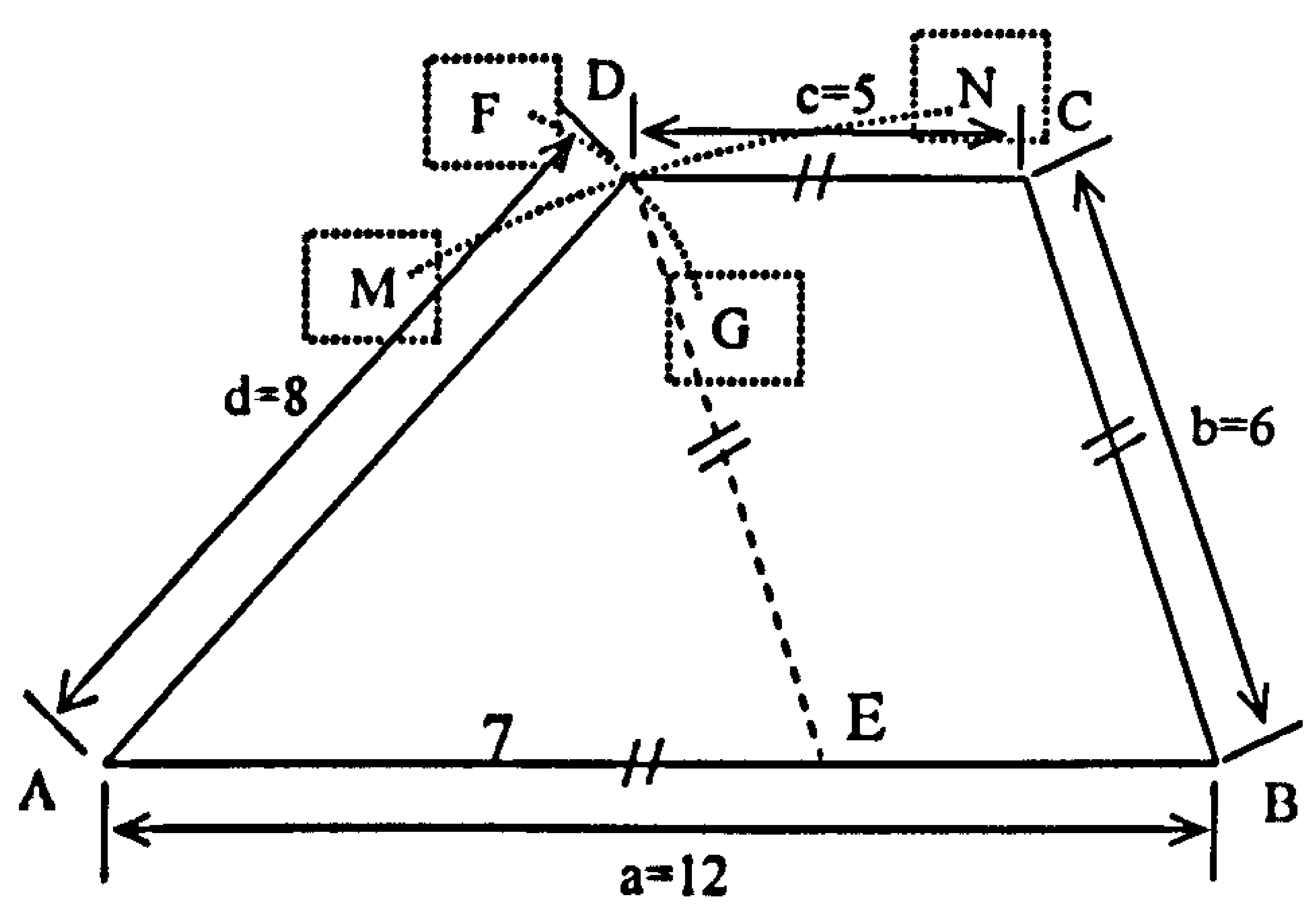


Figure 2.3: Example shape illustrating the idea of sequential geometry modelling

However, this is not the approach adopted by most designers. And it is not clear which parameter is causing the problem when this set of equations yields no solutions. Common practice is that designers tend to specify a set of geometric quantities and topology constraints and then ask the CAD system to generate the geometry. In most cases, these definitions will result in a set of equations, which leads to the requirement for a solver if a general geometric sketching system is to work. It would be more intuitive if a sequential process would be used and information about parameter combinations that fail to produce feasible geometries were available.

To achieve this, the geometry can be modelled with the support of constructive entities introduced in the modelling process. A constructive entity is defined as an entity that is used to transform the implicit rules (defined in form of a set of equations) into explicit rules (modelling process), for example, circles FDG (with the centre at point A and the radius equal to d) and circle MDN (with the centre at point E and the radius equal to b). These two circles are used here to determine the location of point D. The use of constructive entities eliminates the need for a geometric constraints solver here and makes the modelling process become a sequential one. The modelling process for this simple example would then be:

- Determine point E using rules $AE = AB - DC = a - c = 7$;
- Locate point D given the three edges of the triangle ADE via the use of two circles;
- Determine point C using the length of $CD = c$;
- Finish the shape.

The definition of this quadrilateral can be further illustrated by the IDL lisp (ICAD definition language, which is an extension of Common LISP language[41] covering geometry modelling). Each model (called Defpart in the ICAD) can consist of any number of fully defined entities. The basic building blocks in geometry modelling using ICAD are those shape primitives such as points, lines, and circles, etc. and all those shapes already defined by the user. In this example, four straight lines and two circular arcs are used, as shown in the IDL code given in Figure 2.4. The circular arcs are used as constructive entities in assisting the determination of point D. Each Defpart normally contains several sections which include *:inputs* or *:modifiable-optional-inputs*; *:attributes*, *:pseudo-parts*, and *:parts*. The composing entities are usually placed in the *:parts* section and constructive entities are normally placed in the *:pseudo-parts* section and act as supportive entities. Different attributes including geometric ones and non-geometric ones can be included in the *:attributes* section and be referenced when the Defpart form part of a high lever entity.

```
(defpart quad (base-object)

 :modifiable-inputs
 (:center)

 :modifiable-optional-inputs
 (:length-a 12
  :length-b 6
  :length-c 5
  :length-d 8)

 :attributes
 (:radius-fdg (the :length-d)
  :radius-mdn (the :length-b)
  :point-a (the :center)
  :point-b (translate-along-vector (the :point-a)  (the (:face-normal-vector :right))  (the :length-a))
  :point-e (translate-along-vector (the :point-a)  (the (:face-normal-vector :right))
                                    (- (the :length-a) (the :length-c)))
  :point-d (inter-two-planar-circles (the :point-a) (the :radius-fdg) (the :point-e) (the :radius-mdn)
                                      (the (:face-normal-vector :top))  t)
  :point-c (the :line-dc :end)
  :perimeter (+ (the :line-ab :length) (the :line-ad :length) (the :line-dc :length) (the :line-bd :length))
 )
 :pseudo-parts
 ((:circle-fdg :type arc-curve
               :end-angle *pi*
               :arc-constraints
               (:center (the :point-a)
                :radius (the :length-d)
                :start-vector (the (:face-normal-vector :left))
                :plane-normal (the (:face-normal-vector :bottom))))
  (:circle-mdn :type arc-curve
               :end-angle *pi*
               :arc-constraints
               (:center (the :point-e)
                 :radius (the :length-b)
                 :start-vector (the (:face-normal-vector :left))
                 :plane-normal (the (:face-normal-vector :bottom)))))
 :parts
 ((:line-ab :type linear-curve
            :line-constraints
            (:through-point (the :point-a)
             :through-point (the :point-b)))
  (:line-ad :type linear-curve
            :line-constraints
            (:through-point (the :point-a)
             :through-point (the :point-d)))
  (:line-dc :type linear-curve
            :line-constraints
            (:through-point (the :point-d)
              :at-angle (:angle 0
              :reference-vector (the (:face-normal-vector :right))
              :plane-normal (the (:face-normal-vector :top)))))
  (:line-bc :type linear-curve
            :line-constraints
            (:through-point (the :point-b)
             :through-point (the :point-c)))))
```

Figure 2.4: IDL code for quadrilateral geometry

It is obvious that some combinations of these four parameters (the lengths of four edges a,b,c,d) will lead to incompatible geometries. However, these parameters are still independent of each other. Any of the four can be independently varied while holding the other three constant, though the other three affects its range. If the parameters can be categorized into primary and secondary, then the range for the secondary parameters can be obtained using the geometric rules based on the values of primary parameters, thus limiting the search space. If no priority can be predetermined, relatively large ranges should be tried to avoid missing any area in the search space. This may lead to large number of unusable geometries when Design of Experiment (DoE) methods are used to prepare dataset for surrogating modelling, as typical DoE methods attempt to uniformly cover the search space. A sequential method can be effectively used to check the usability of the geometry and make corrections as such that a set of usable geometries can be obtained. The method is based on sequential approach for geometry modelling. Rules used in determining the range within which a feasible geometry can be obtained can also be used to check the feasibility of geometry determined by a set of given parameters. Although it may be the case that different parameterisations may result in different behaviour of the model, thus affecting the results of optimisation, it is not always possible to ensure usable geometries being produced without an appropriate enforcement of the range of variations of shape parameters. Furthermore, the choice of parameterisation also depends on other factors, most noticeably the manufacturability. Therefore, a method of detecting and/or repairing unusable geometries is required. The sequential modelling approach is described next with application on turbine blade firtrees, followed by the procedure for feasibility checking.

A parameterised geometry shape is determined by a set of independent parameters. Although these parameters are independent to each other, the valid range of one parameter may be linked to the values of other parameter(s). In most engineering optimisation problems, bound constraints are normally applied to the design variables, which are specified by the designer, either based on experience, or physical knowledge. However, it is sometimes difficult to specify suitable ranges for the design variables without the use of problem specific knowledge. This range definition is important as it fixes the search space for the optimiser to explore. When it comes to the use of design of experiments tools to generate data sites for physical experiments or computer simulation, it becomes even more important, as most design of experiment methods use the hypercube formed by the bound values as a reference box to sample the search space. A suitable hypercube is always desired for efficient and effective data sampling. In the case of a parameterised geometry, a large number of infeasible geometries may result when an inappropriate hypercube is used.

2.2.2. **Firtree geometry with single-arc fillet**

Defining a firtree geometry requires a number of parameters, and it can be imagined that there exist many different types of definition methods. For example, in the work reported by Lee *et al.*,[25] the firtree was defined using several angular and straight dimensions, as shown in Figure 2.5. This type of definition is probably the most widely used in the literature. However, it is not possible to produce different size teeth using this approach. And change to one parameter will produce effects on all the teeth. This approach is not compatible to the widely accepted concepts and practices used for object-oriented design methodology in software engineering, which have proved to be performance robust and cost effective in the software industry. Here, therefore, attention is focused on designing a basic tooth object that can be reused.

R = Radius of lobe in blade or in disc
d = lobe shear plane length
c = Load flat contact length projection
θ = Load flat angle
α = Unload flat angle
β = Wedge angle

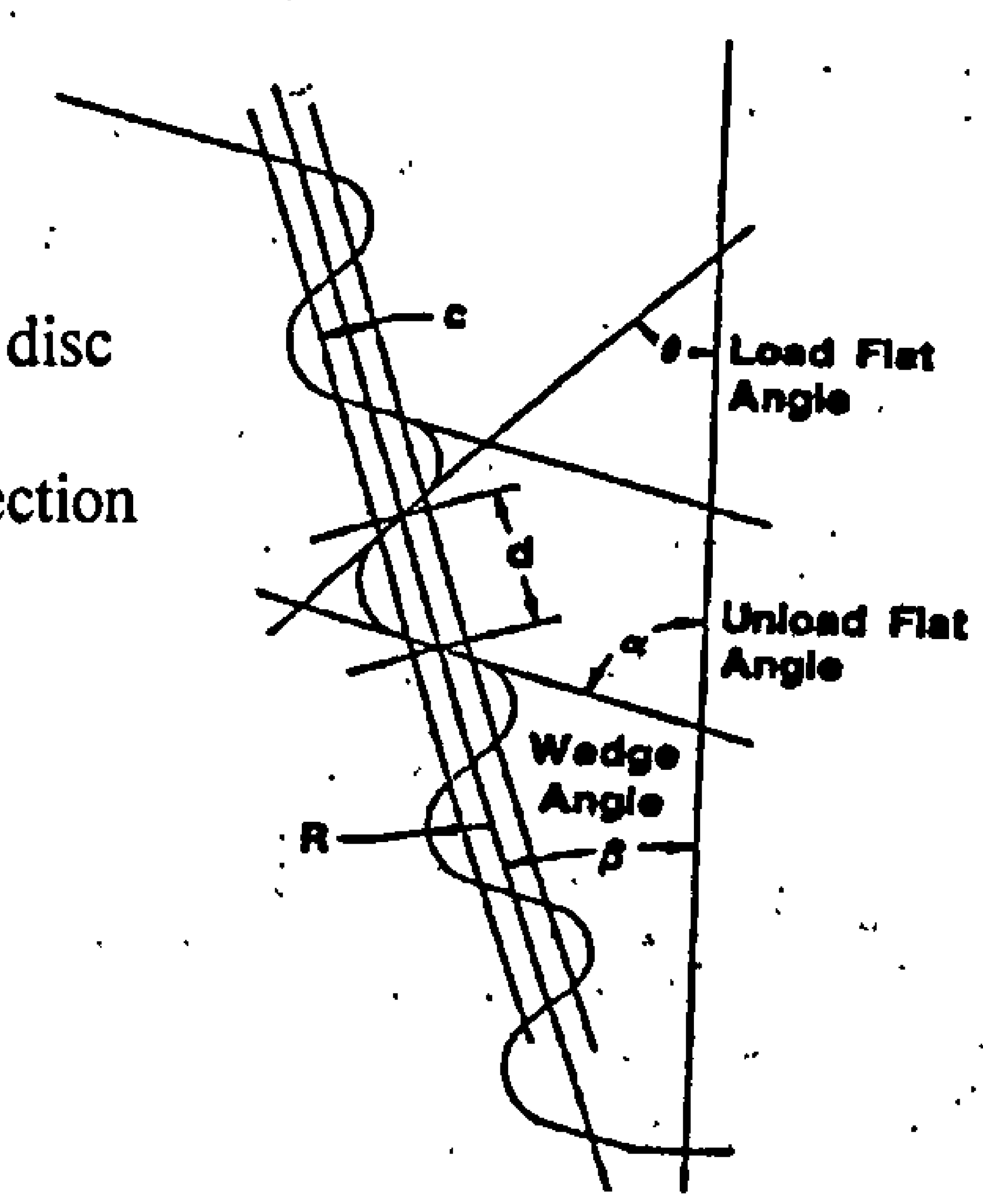


Figure 2.5: A typical definition of firtree geometry usually found in literature

- **Definition of base tooth with single arc**

The single basic tooth model is illustrated in Figure 2.6, which is defined in such a way as to allow designers to explicitly control the non-contact clearance and to avoid duplicate entities in the model, which means that there is only one entity for the common part of blade and disc instead of two overlapped entities. This latter feature eases the application of boundary conditions and loadings during the following analysis phase. This single-arc tooth profile can be defined using ten parameters in total as shown in Figure 2.6; however, some of these parameters such as *root-wedge-angle*, *non-contact-face-clearance*, *top-flank-angle*, *under-flank-angle*, and *next-top-flank-angle* can be shared by all the teeth for a typical design adopted in engineering practice. The connectivity and continuity between two teeth is maintained by starting the next teeth from the end-point of the previous tooth and sharing the value of the parameter *next-top-flank-angle* with the *top-flank-angle* of the next tooth. Therefore the total number of independent variables will be (5+5×number-of-teeth). However, this definition will provide more flexibility than the one shown in Figure 2.5 as parameters take different values for different teeth.

Geometry of the basic tooth is defined using the sequential method described in the previous section. The detailed sequence of definition is illustrated in Figure 2.7, and a section of IDL code is included in Appendix A, in which, the detailed structure of firtree geometry model is also given. It needs to be pointed out that although various parameterisations are possible, just as one shown in Figure 2.5, which is widely used in the literature. A balance between flexibility and consistency with practical experience needs to be considered. In addition, manufacturability and cost issues also have a role to play. Clearly, the more information used, the more effective the parameterisation will be. And experiences with the interplay between parameterisation and optimisation will help improve the parameterisation itself, as parameterisation essentially limits the size and shape of the search space of the problem.

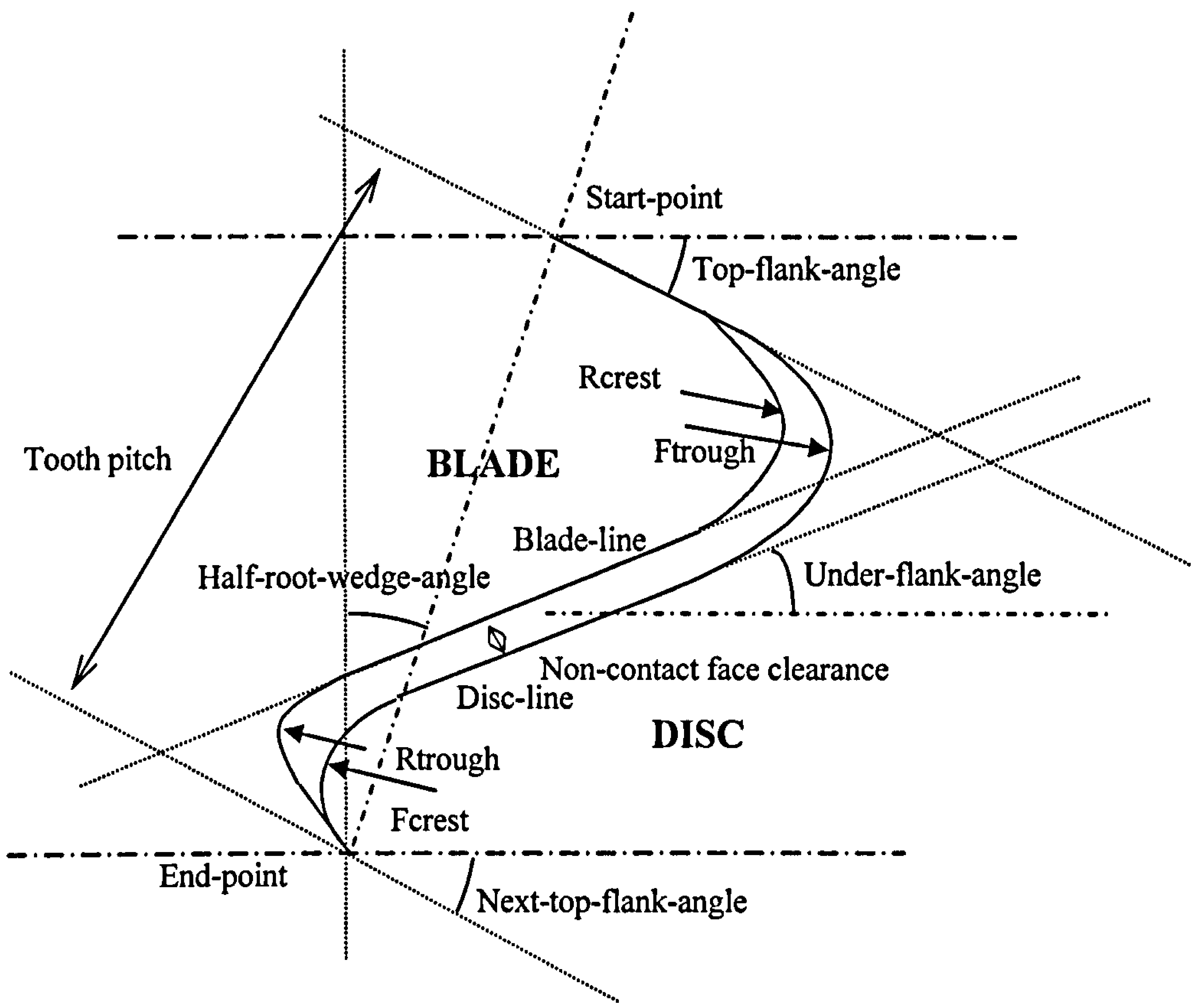


Figure 2.6: Geometry definition of a basic tooth

- Determine *end-point* given *start-point* (reference point), *tooth-pitch, half-root-wedge-angle*, and *top-flank-angle;*
- Draw circular arc disc-crest using given radius *Fcrest* and two angles: *under-flank-angle* and *next-top-flank-angle;*
- Draw disc line tangent to disc crest;
- Draw blade line parallel to the disc line and with a gap between them equal to *non-contact-face-clearance;*
- Draw the three remaining fillets blade crest, blade trough and disc trough using given radius: *Rcrest, Rtrough,* and *Ftrough.*

Figure 2.7 Sequential modelling process of the basic tooth

- **Firtree definition with single arc tooth**

Having defined a basic tooth profile, the blade root and disc head geometry can then be defined in the similar manner as the basic tooth, with further parameters and rules being added. The first firtree geometry being tackled is derived from an existing firtree model, which is composed of straight lines and circular arcs only. The complete geometry is described by approximately 26 quantities (with the blade tooth pitch and disc tooth pitch always identical), as shown in Table 2.1. Some of the parameters used in the definition may be determined from other sources, therefore are not expected to change. These parameters are thus held constant during optimisation, while others that are identified as being more influential to the design will be varied by the optimiser and are known as design variables.[42] Some parameter, for example, the number of teeth, can either be kept constant or varied by the optimiser depending on the target of the particular search. In an attempt to fine-tune an existing design, number of teeth is usually not expected to change. However, if the objective of the search is trying to discover possible radically new design, as many as possible parameters should be allowed to change. In this case, parameter screening and other methods can be used to identity the rank of importance of the parameters.[43] The resultant blade/disc geometry is illustrated in Figures 2.1 and 2.2 presented earlier.

2.2.3. Computation of geometric and non-geometric properties

Every entity within the ICAD model can have other non-geometric properties as well as its inherent geometric properties. Examples of these properties include tag name, material type, mesh density, etc. The use of these properties eases the use of the geometry in other applications such as analysis and manufacture. For example, to apply boundary conditions and loads to entities during the analysis stage using the finite element method, it is desirable to name the entities with unique labels that can then be referenced at later stages of the design process. Using unique tag names on each entity in the geometry enables us to specify the boundary conditions, load properties and mesh properties in batch mode, which is essential to automate the whole process. The tag names are transferred to the finite element code along with the geometry using the geometry standard IGES,[26] which supports the naming of entities within the geometry.

The specification of boundary conditions and loads relies on the facilities provided by the analysis code.

Other derived geometric quantities such as the minimum thickness of the blade root when a cooling passage is considered in the model, distance between the centre of contact face of the tooth on each side, etc., are calculated in the ICAD model based on the mathematical representation of the geometry. Some of these are treated as constraints in the optimisation problem and some are used to generate the FE model or to retrieve the analysis results. For example, point coordinates are normally required to get the stress values at those points. The bedding length is required to calculate the crushing stress.

Apart from additional non-geometric properties that can be included in the geometry model, the following benefits can be achieved using the concepts of object-oriented design and version control facilities available in ICAD:

- Multi-modality: the objects can have different internal implementations, while the same interface can be maintained; this allows upgrade or improvement in the implementation while maintaining the interface.
- Inheritance: objects which implement the basic protocols can be used in the implementation of its sub-classes, and thus constitute a hierarchy structure of product. Properties can be made accessible to other objects.

Table 2.1 Blade root/Disc head geometry parameters and base design

Variables	Names	Values	Units	Type†
	Blade Tooth Profile Parameters			
Snw	Shank-neck-width	8.0	mm	Variable
Rcrest	Blade-tooth-crest-radius	0.5461	mm	Variable
Rtrough	Blade-tooth-trough-radius	0.4699	mm	Variable
Bp1, 2	Blade-tooth-pitch	3.3038	mm	Variable
Btcr	Bottom-tooth-crest-radius	1.0668	mm	Variable
Cpw	Cooling-passage-width	1.39296	mm	Variable
	Disc Tooth Profile Parameters			
Dtp1, 2	Disc-tooth-pitch	Bp1,2	mm	Variable
Fcrest	Disc-tooth-crest-radius	0.5969	mm	Variable
Ftrough	Disc-tooth-trough-radius	0.4191	mm	Variable
Bglr(R1)	Bucket-groove-lower-radius	3.5	mm	Variable
Bgur(R2)	Bucket-groove-upper-radius	2.2	mm	Variable
Rtsn	Radius-to-shank-neck	213.225	mm	Constant
	Common Firtree Parameters			
Nteeth	Number-of-teeth	3		Variable/Const.
Skew(β)	Root-skew-angle	15	degree	Variable
Rwa	Root-wedge-angle	31.8428	degree	Variable
Fsw	Firtree-shoulder-width	10.2433	mm	Variable
Alor(L)	Axial-length-of-root	20	mm	Variable
Bac(Ca)	Blade-axial-chord		mm	Constant
Ncfc	Non-contact-face-clearance	0.0971	mm	Constant
Snfr(R)	Shank-neck-fillet-radius	2.0	mm	Constant
Tfa(ϕ)	Top-flank-angle	47.5	degree	Constant
Ufa(γ)	Under-flank-angle	22.5	degree	Constant
	Disc Parameters			
Nblades	Number-of-blades	54		Constant
Drad	Disc-radius	212.715	mm	Constant
Fdcr	First-disc-crest-radius	1.0	mm	Constant
Ninc	Number-of-blades-inclusive	1/3		Constant
Inr	Inner-radius	62.0635	mm	Constant

†Notes: Type of a parameter indicates that whether the parameter is held constant or being varied in optimisation.

2.2.4. Firtree geometry with double-arc fillet

Although the single arc is the widely adopted choice for fillet geometry, it has apparent weakness in reducing the stress concentration. A straightforward extension to the single arc is to link more sections of arcs together to form a multi-arc fillet, among which, the double arc fillet is the easiest to define.

Given two lines and two fillet radii, an extra parameter, called θ (as shown in Figure 2.8), is required to determine the configuration of composite fillet geometry. Details on the modelling process are described in the following paragraph in the form of relations between geometric quantities, and constraints on parameters.

Given: R_1, R_2, and θ, the following relations hold from the two adjacent quadrilaterals $ACEH$ and $GEDB$, respectively

$$AC = CE = R_1 * \tan(\theta/2) \tag{2.2.4.1}$$

$$BD = ED = R_2 * c\tan(\frac{\alpha+\theta}{2}) \tag{2.2.4.2}$$

Combining the above two equations, and using the sine formula, we have

$$CD = CE + ED = R_1 * \tan(\theta/2) + R_2 * c\tan(\frac{\alpha+\theta}{2}) \tag{2.2.4.3}$$

$$OD = CD * \sin\theta / \sin\alpha \tag{2.2.4.4}$$

$$OC = CD * \sin(\alpha+\theta) / \sin\alpha \tag{2.2.4.5}$$

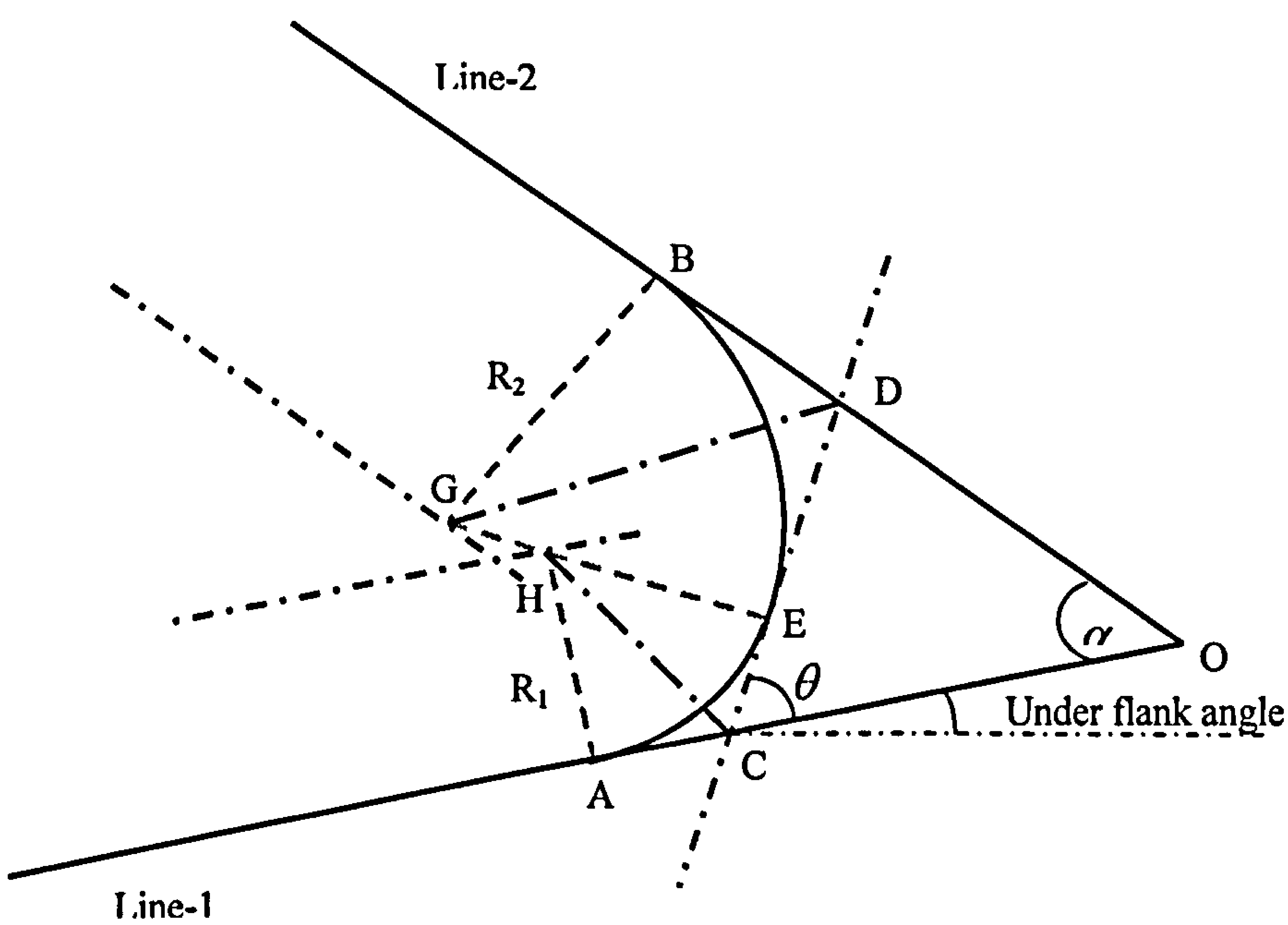


Figure 2.8: Illustration of double-arc fillet geometry

Therefore, the points C and D can be determined by the dimensions of OC and OD. Two arcs will then be determined by the centre points G, H and two radii specified by the user. A single tooth profile with double-arc fillet can be defined using the geometric rules derived above, as shown in Figure 2.9. Parameters that differ from the single arc tooth are defined in Table 2.2. The extra parameter θ acts as a partition parameter between the two circular arcs and can be fixed in order to reduce the number of parameters. For example, the value of $\theta = \pi/2 - under\_flank\_angle$ will produce a vertical line tangent to both circular arcs.

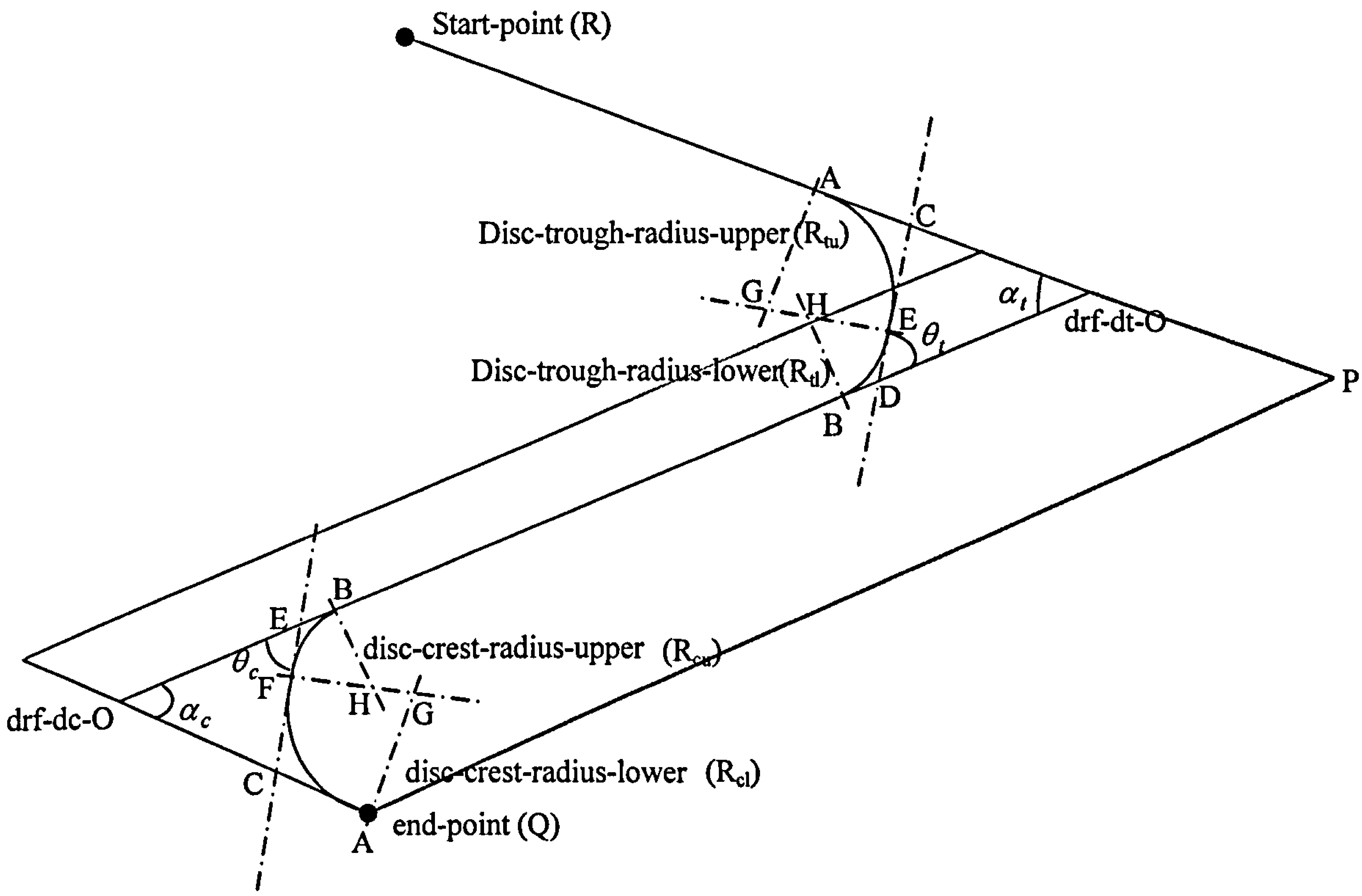


Figure 2.9: Geometry of A single tooth profile with double-arc fillet

Table 2.2 Firtree geometry parameters and base values for double-arc fillet

Variable	Name	Value	Units	Type
	Blade Tooth Profile Parameters			
Bp1, 2.	Blade-tooth-pitch	3.3038	mm	Variable
Rctau/l	Blade-tooth-crest-radius-upper/lower	0.91/0.46	mm	Variable
Rtrtu/l	Blade-tooth-trough-radius-upper/lower	0.46/0.91	mm	Variable
Btau/l	Bottom-tooth-crest-radius-upper/lower	0.91/0.46/	mm	Variable
	Disc Tooth Profile Parameters			
Dtp1, 2.	Disc-tooth-pitch	3.3038	mm	Variable
fctau/l	Disc-tooth-crest-radius-upper/lower	0.46/0.91/	mm	Variable
ftrau/l	Disc-tooth-trough-radius-upper/lower	0.91/0.46	mm	Variable

The relations between fillet radii for the double-arc profile become more complex than single-arc profile. However it is still possible to derive the relations in the form of inequality constraints for the radius parameters.

Take the disc tooth profile as an example, the following geometric inequality constraints need to be satisfied to produce compatible tooth profiles:

$$OA_{drf-dt} + OA_{drf-dc} \leq RP \tag{2.2.4.6}$$

$$OB_{drf-dt} + OB_{drf-dc} \leq PQ \tag{2.2.4.7}$$

These relations can be expanded using the fillet radii to the following linear form, making use of the geometric rules represented by equations (2.2.4.1~2.2.4.5):

$$\beta_{11}R_{tu} + \beta_{21}R_{tl} + \beta_{31}R_{cu} + \beta_{41}R_{cl} \leq RP \tag{2.2.4.8}$$

$$\beta_{12}R_{tu} + \beta_{22}R_{tl} + \beta_{32}R_{cu} + \beta_{42}R_{cl} \leq PQ \tag{2.2.4.9}$$

where $\beta_{ij},(i=1,...4, j=1,2)$ are coefficients determined by the four angular parameters $\alpha_c, \alpha_t, \theta_c, \theta_t$ These linear inequality constraints can be used in implementing the geometry compatibility check and determining the combinations of fillet radii to produce compatible geometries.

2.3. Identification and automatic repair of geometry problems

Geometry related problems account for many of the difficulties encountered in high-fidelity numerical analysis programs, such as finite element methods and computational fluid dynamics. Some repair facilities are normally available in commercial analysis codes to tackle this type of problem. The geometries produced by CAD systems are often imported into the high-fidelity code as a starting point for analysis, and immediately following are usually topology and geometry repair processes, as for example in ANSYS. However, these manual or semi-automatic methods are not applicable in situations where a complete automation is required. Generating a robust geometry is always more desirable than using a repaired one.

The usability of any firtree geometry needs to be checked since some particular combination of parameters may result in unusable features such as intersections between two entities or presence of very short entities. The handling of unusable features is important to the optimisation process as well as to the analysis code. Using ICAD, any unusable geometries can be checked within the modelling process, and appropriate actions can then be taken, such as using the nearest possible values, or simply signalling the analysis code to cancel the analysis.

The following sections describe the rules used in the usability check process. This check is performed in a sequential manner, so whenever the check fails to produce usable geometry, the firtree status is set to 1. The status value is returned to indicate unusable geometry. Various problems can occur in the geometry modelling process. An example of unusable geometry is shown in Figure 2.10, in which, intersection occurs between the blade trough and disc crest.

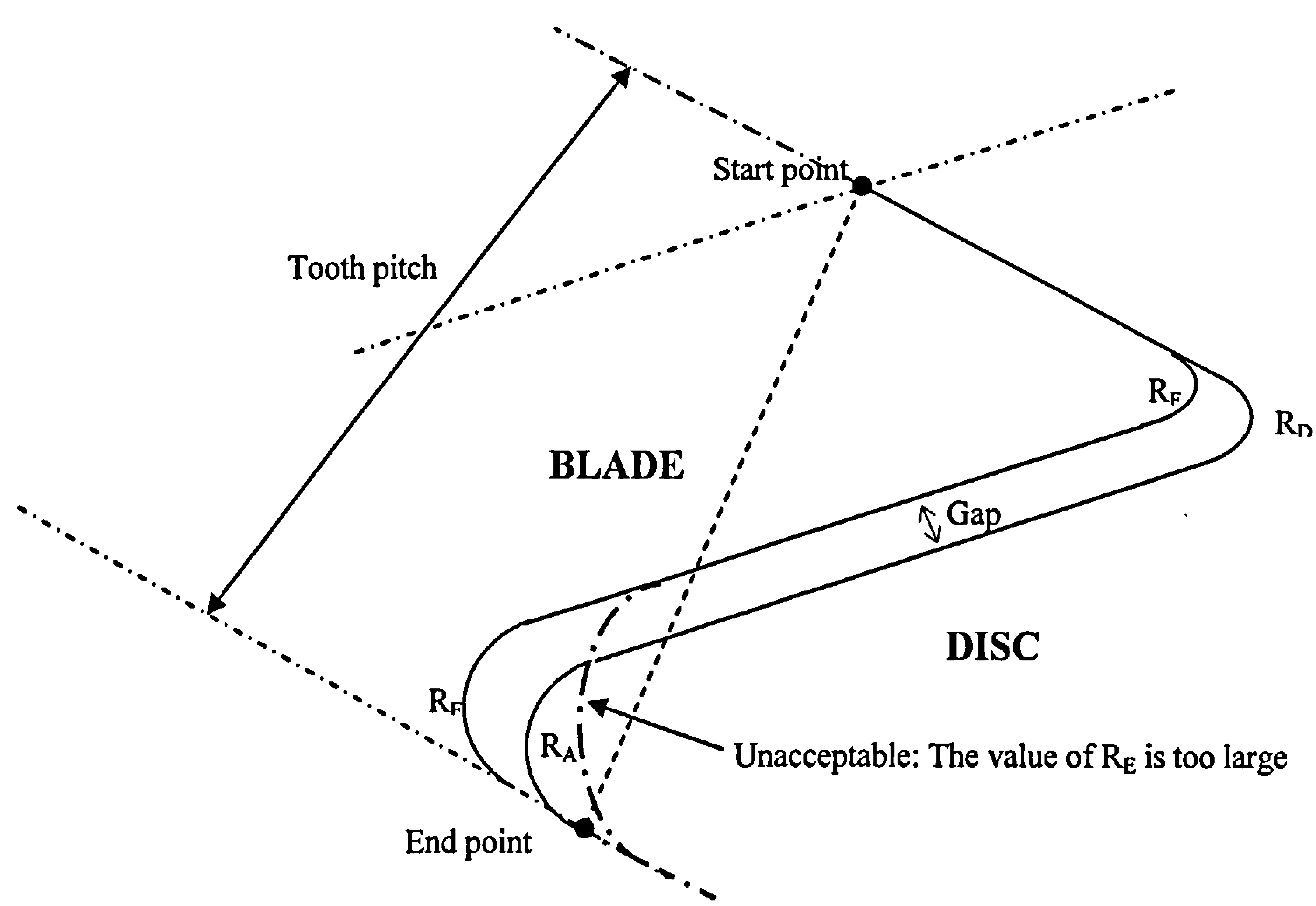


Figure 2.10: A base tooth and an example of unacceptable geometry with end point fixed

The following rules are applied in the firtree geometry modelling process.

- (i) Cooling passage width should be less than the shank neck width (snw), see Figure 2.11a;
- (ii) The lower limit on the shank shoulder width (fsw) is determined by two parameters, shank neck width and shank neck fillet radius, see Figure 2.11a;
- (iii) The minimum wall thickness should be greater than zero, see Figure 2.11a;
- (iv) The minimum distance to the two sides of disc section should be greater than zero, see Figure 2.11a;
- (v) The upper limit on the bottom tooth crest radius is determined when the length of bottom line is reduced to zero, see Figure 2.11b;
- (vi) R1 on bucket groove region should be greater than bottom tooth crest radius, see Figure 2.11b;
- (vii) Each tooth is bounded in a rectangular box, see Figure 2.11c.

The starting point for the first tooth is determined by the shank-shoulder-width, and the starting point for the following tooth is the end point of the previous tooth; The end point of each tooth is determined by the tooth pitch and half root wedge angle; Within the bounding box for each tooth, the entities are determined in the following order: *disc-crest*, *disc-line*, *firtree-trough*, *disc-trough*, *firtree-line*, *firtree-crest*, and *bedding-line*; The values of parameters defining these entities are checked against the lower and upper bounds prescribed by existing entities, this forms a sequential modelling process, as shown in Figure 2.11d.

There are two optional actions that can be taken when an unusable geometry would have been produced for a particular set of parameters. During the search process, this can be signalled to the analysis code to cancel the analysis and assign a very large value to the objective function (for minimization problems). This is easy to implement and also effective when coupled with most optimisation methods. However, this method would severely distort the response surface constructed from design of experiment results when a large number of design points in a Design of Experiment correspond to unusable geometries. The reason is because the dataset is saturated with designs with no sensible function values. The second option in case of an unusable geometry is to assign a default value to the variable that causes the problem. As the modelling process is essentially a sequential one, the variable that causes the problem can be identified and a default value can be assigned to it. In this way, the number of data point with no sensible function values.

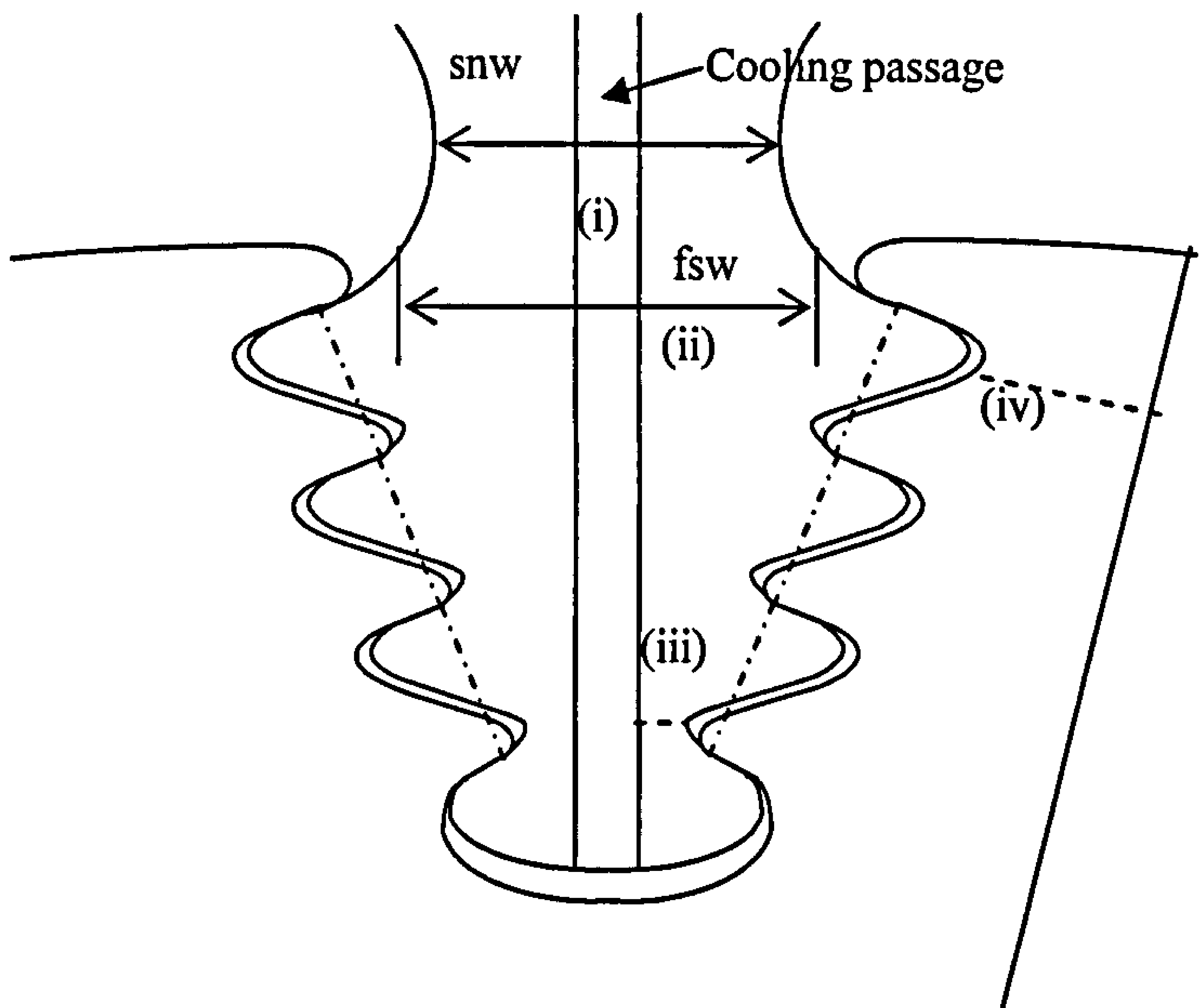


Figure 2.11a: Quantities used for rules (i), (ii), (iii), (iv)

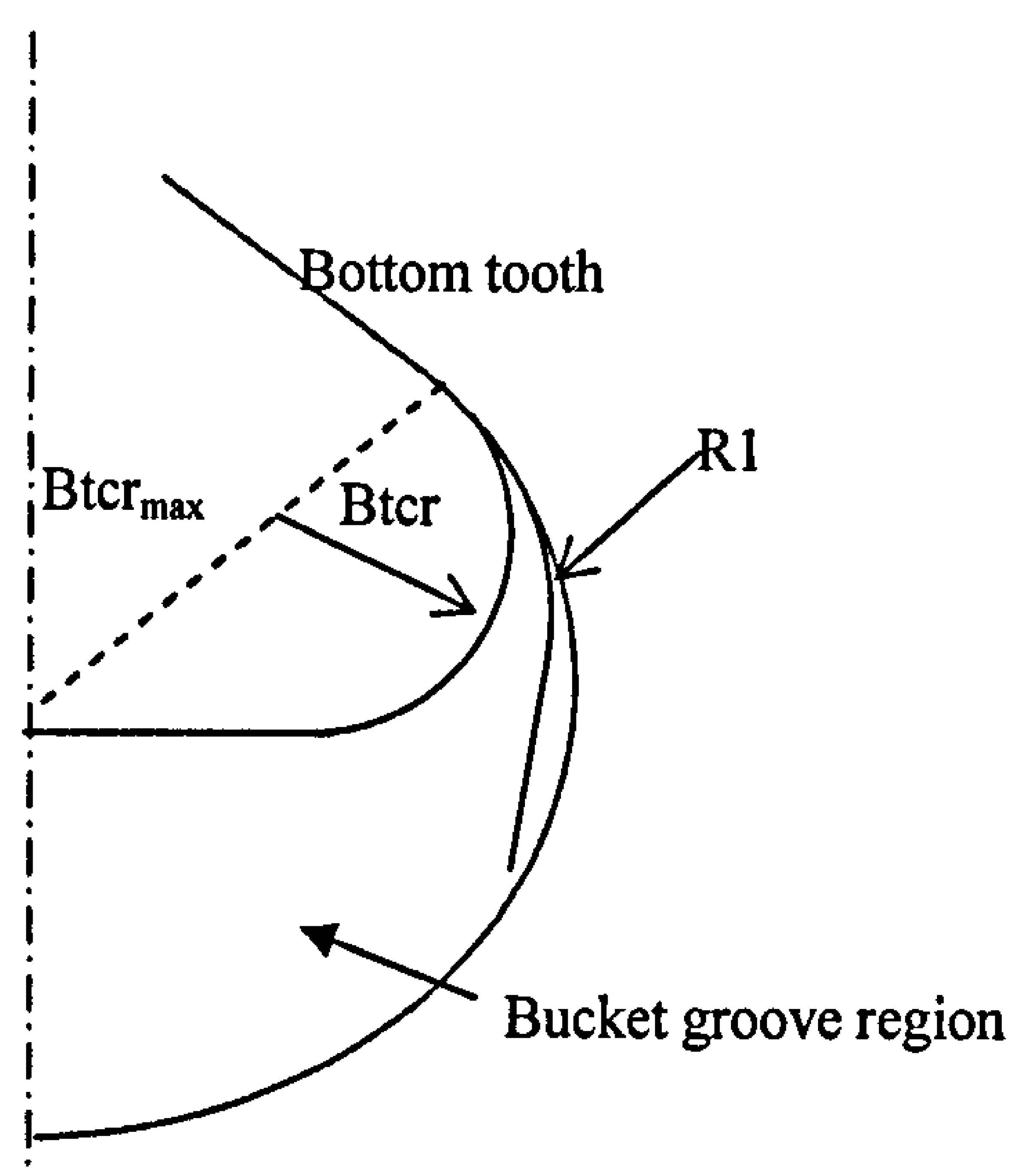


Figure.2.11b: Quantities used for rule (v) (vi)

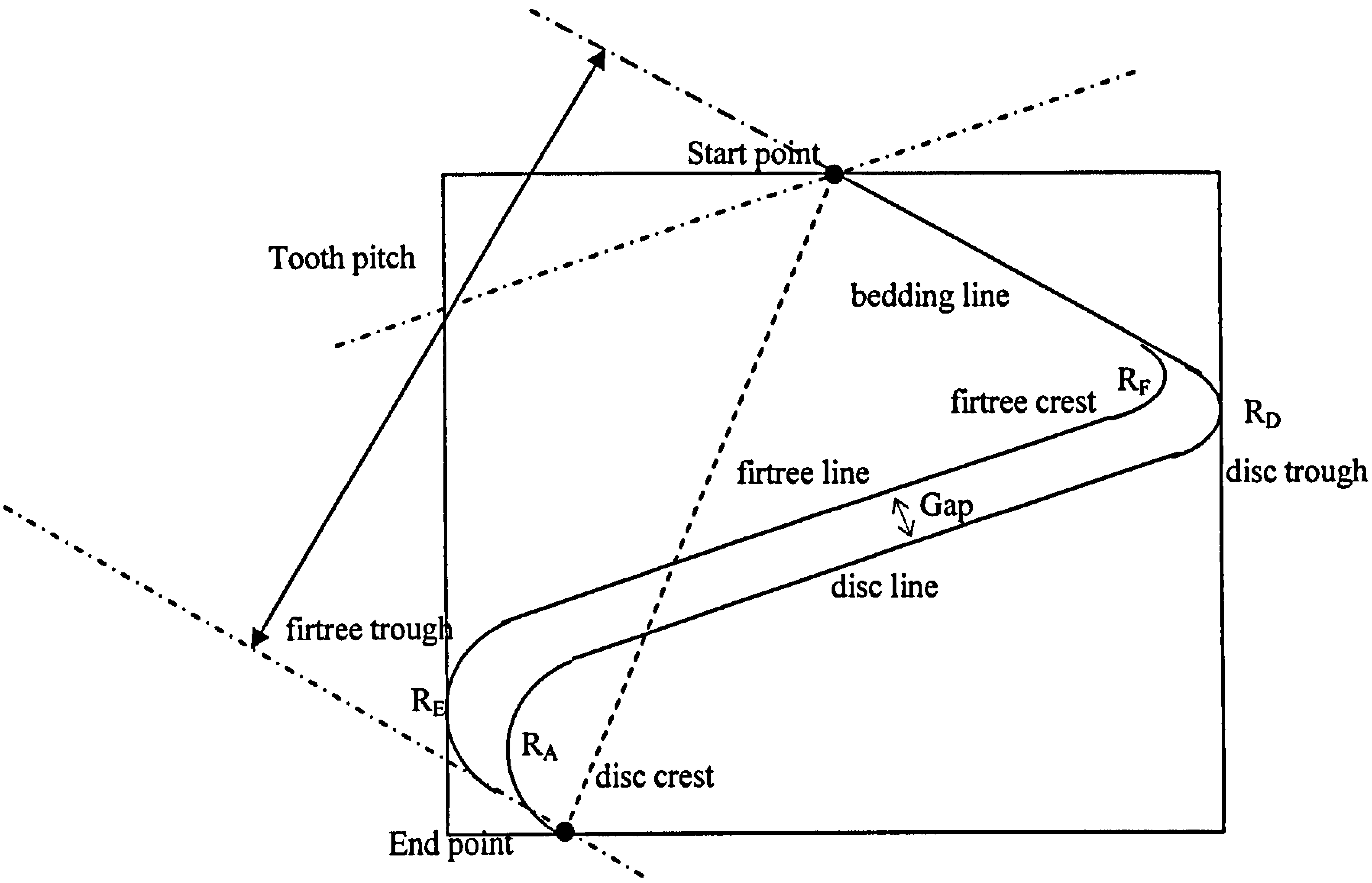


Figure 2.11c: Base tooth geometry and boundary box (rule vii)

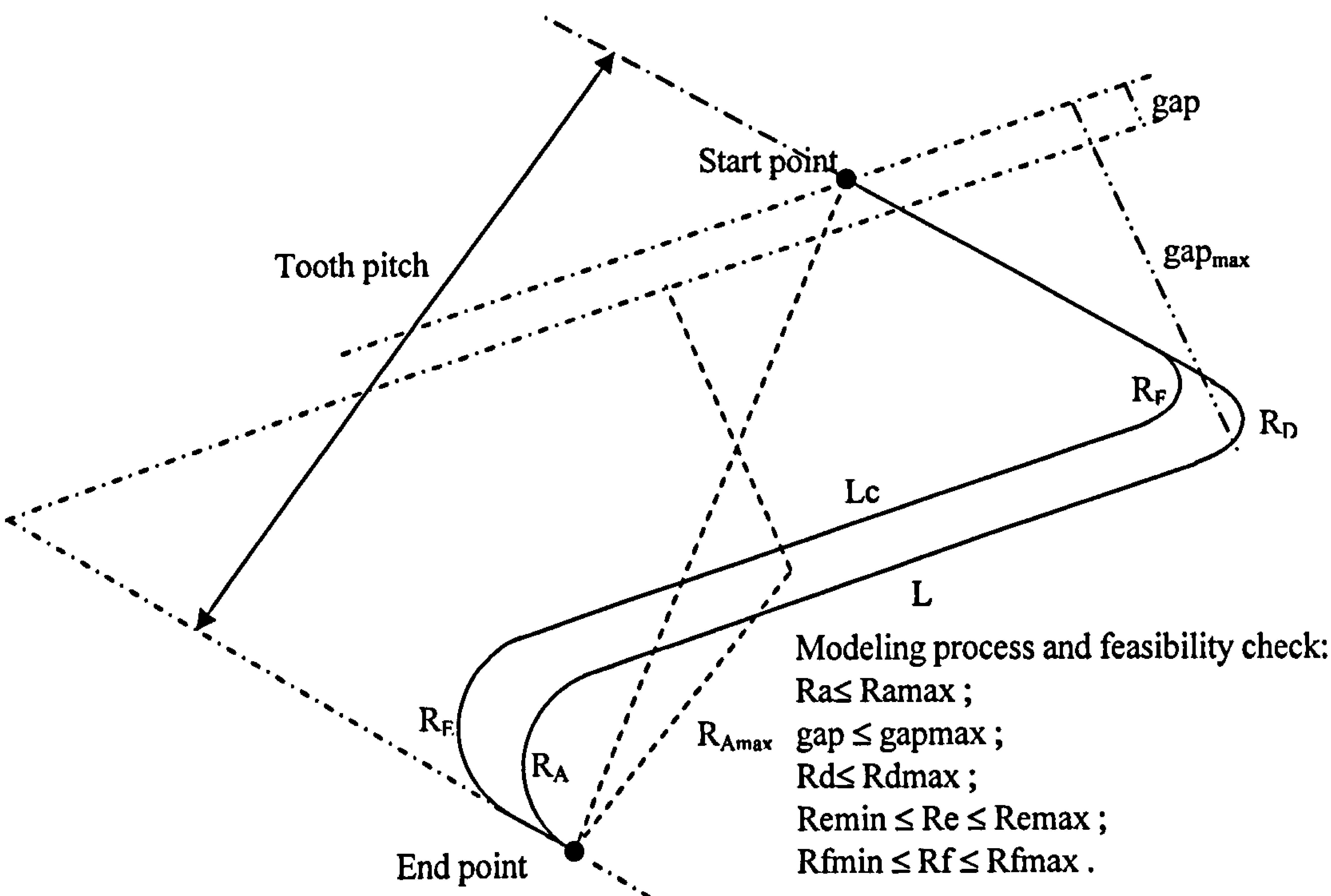


Figure 2.11d: Feasibility check process of single tooth profile

Chapter 3

Finite element modelling of firtrees

3.1. Introduction

The firtree like attachment used to position a blade airfoil at the proper radius in a turbine structure is usually identified as a critical area in an aero-engine design. Dovetail is another form geometry usually used in some type of fan or compressor blade/disc configurations. The component is subject to high mechanical loads produced by the high-speed rotation of the blade. Most often the attachment is a multi-lobe construction used to transfer the loads from blade to disc. It is generally assumed that there are two forms of loading which act on the firtree, the primary radial centrifugal tensile load resulting from the rotation of the disc, and bending of the blade as a cantilever which is produced by the combined action of the gas pressure on the airfoil and forces due to tilting of the airfoil. The resulting stress distribution in the root attachment area is a function of geometry, material properties and loading conditions, which are of course related to the speed of rotation. Although several stress quantities like section average stresses can be estimated using simple equations derived on the assumptions of uniform division of centrifugal load between all teeth,[22] detailed stress distribution in the firtree cannot be confidently obtained without the extensive use of finite element methods, especially when complex friction mechanisms are included in the analysis.

A number of studies into the stress state of the blade root attachment have been reported, originally using photoelastic methods, for example, see by Durelli *et al.*,[23] or using finite element analysis, see Refs. 44-47. The stress analysis of dovetail or firtree is not without challenge. And the analysis of single tooth dovetail or multi-lobe firtrees is further complicated by the fact that there is usually complex frictional interactions between the two bodies. Previously published work mainly deals with relatively simple geometries and the aims of these studies are mainly for investigating the stress properties and effect of some geometric features. Boddington *et al.*[44] considered different states of the interface between the two bodies by using different boundary conditions for each mesh at the common edge. The effect of friction was also included in the model proposed by Boddington. The effect of the shape imperfections on the stress state was investigated by Dibsky[45] by considering two extreme states – upper point contact and lower point contact. A three-dimensional finite element analysis was carried out by Papanikos *et al.*[46] in order to address the inadequacy of two-dimensional analysis and provide a good presentation with regard to the differences between two and three-dimensional models.

Different aspects of the challenges in the stress analysis when using finite element methods were identified by Sinclair *et al.*,[47] which include large stress gradient around the edge of contact, geometric nonlinearity associated with expanding contact regions when contact is conforming, and tackling the effect of frictions.

Analytical solutions have also been attempted for the simple dovetail geometry in order to gain some physical insight into the problem, see Sinclair *et al.*[48] The simple physical models are useful in understanding the effect of various factors on stress concentration and provide useful ways to identify possible failure mechanisms. However, it is usually too complex to derive sound physical models for complex firtree shapes. And the results from physical models also invariably need to be verified by detailed finite element analysis.

Although three-dimensional stress analysis of a firtree using finite element analysis can provide accurate stress estimates, the computational cost associated with it may be too high to be used in optimisation, and construction of a three-dimensional model is also time-consuming and prone to problems. Therefore, it is not sensible to incorporate a three-dimensional analysis at the outset. A two-dimensional model can be used with considerations of three-dimensional effect using scale factors called skew factors, which can be estimated from a few three-dimensional analyses.

A number of assumptions proposed by O'Connor *et al.*[49] are widely used concerning the two-dimensional stress analysis of the firtrees:

- The possible contact zones are clearly dented and smooth in shape allowing relatively straightforward matching of the mesh models on two bodies;
- The magnitude of interface motion in the contact region is small;
- The deformations are elastic and gross distortions of the interface unlikely;
- The coefficient of friction is infinite, i.e., there is no consideration of friction between the two bodies.

Based on the above assumptions, several models have been applied in the analysis of dovetail joints. The simplest model assumes a continuous solid model at the beginning, then uses an iterative approach to find out the normal forces along the interface, where the analysis in the following iteration is based on new boundary conditions obtained from previous calculations. The results of this simple method show that very high shear forces exist along the interface. A more sophisticated approach has been developed by O'Connor,[49] in which a matched mesh was used along the interface boundary. This method has been widely used in the analysis of contact problems, for example, see Boddington.[44] One alternative would be using non-matching meshes in the boundary region. However this approach showed mixed agreement with experimental observations while further complicating the situation. Special boundary conditions have to be formulated to balance the discretized forces and displacements of the two bodies.

The finite element model used here is a node-to-node two-dimensional model with infinite friction coefficient. The same discretization is used for the two bodies on the common edge. The analysis starts with an initial assumption of the node-pair displacements, then after each iteration, node-pair consistency is checked, i.e., there should be no overlap for open node-pairs and no tensile forces for closed node-pairs.

3.2. Finite element model of firtree root

It has been shown from previous preliminary optimisation work that it is possible to reduce the number of teeth in a firtree joint.[50] From aerodynamics consideration, a constant Space/Chord ratio ensures a constant lift coefficient, and it is thus possible to trade blade number for blade chord. The reduction of blade number and consequent increase in blade size will cause an increase of centrifugal load transferred to the firtree root. The root will have to be redesigned to cope with the larger load. Fortunately, a reduced number of blades leaves more disc material between adjacent firtree root slots, and the wider disc post thus allows a bigger tooth design to cope with the increased load and this makes a further reduction of firtree teeth numbers viable. The reduced number of teeth will bring further cost savings.

As already noted, the loading on the root is mainly due to centrifugal load which is dependent on the mass of the whole blade which incorporates the root mass. The design of the firtree thus involves an iterative process of the calculation of the total blade mass followed by firtree analysis. Also some key features, such as the selection of fillet radius, play very important roles in minimizing the stresses in notch regions. Thus a set of competitive constraints ranging from geometrical, mechanical, cooling requirements, etc., is established for use in exploration of various design candidates for the firtree root. Finite element analysis is then utilized to obtain the resulting stress distributions. This further complicates the situation. A traditional manual method is now considered too slow and thus automation is required.

Due to the cyclic symmetry of the model, only one sector of the disc and attached blade firtree root is modelled here using the Rolls-Royce in-house finite element code SC03. The cyclic symmetric boundary conditions are applied to the two sides of the disc sector. The primary centrifugal load generated by the blade part above the firtree root at the maximum speed is applied on the shank neck in the form of concentrated load. A moment is also applied on the shank neck as a result of bending of the blade produced by the aerodynamic force on the blade. The centrifugal load generated by the firtree itself and disc section is applied in the form of body forces by applying a rotational speed. Two important aspects of the model lie in the specification of cyclic boundary condition and modelling of contact regions between blade and disc. The cyclic boundary condition will essentially generate a matched mesh on the two sides of the disc sector. It can be seen that asymmetric loading exists which will lead to asymmetric stress distribution, and therefore an asymmetric geometry may provide benefit in terms of stress reduction. However, the benefit may not be great as the asymmetric portion (f_m) of the total load is less than 3 percent of the symmetric load (F_y) on the shank neck, this can be seen from Figure 3.1, in which M_z is considered as the result of application of f_m on two ends of the shank neck. In addition the manufacture cost of an asymmetric geometry need also be considered before any persuasive conclusions can be made.

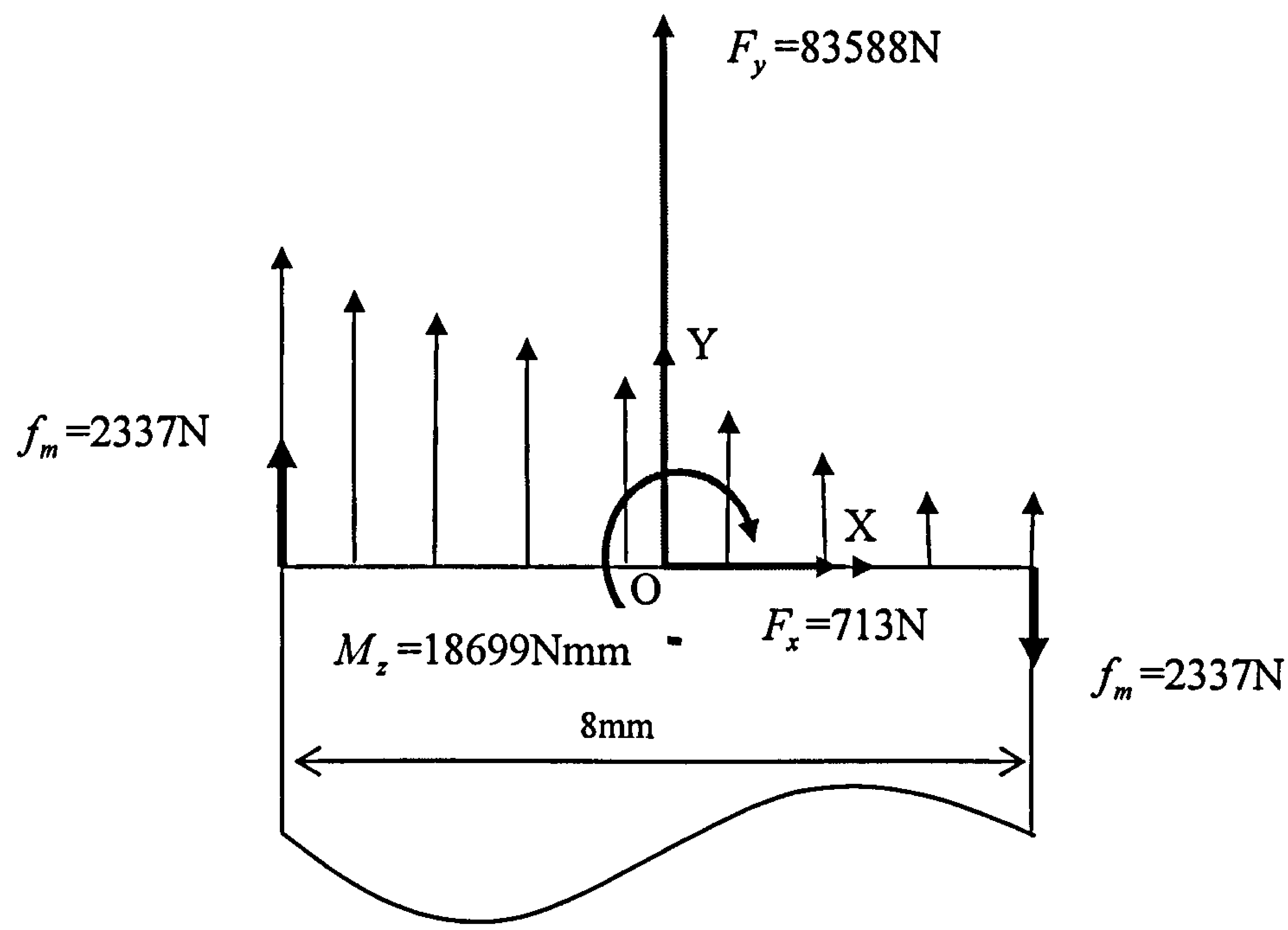


Figure 3.1: Illustration of loading on the shank neck of turbine blade firtrees

The procedure of the finite element analysis is listed below:

- Import the geometry from ICAD system via the use of an IGES file;
- Specify the boundary conditions, material properties, and loading;
- Mesh the geometry;
- Solve the problem and retrieve the results.

Most FE codes support batch running of the analysis and this allows the analysis to be embedded into an overall optimisation loop. Smoothly coupling the geometry modelling with the analysis tool, however, is not an easy task. It involves the transfer of the geometry itself and related geometry dependent properties to the analysis code, in this case, the finite element software. In this work, a piece of FORTRAN code, termed as SC03 plugin, was developed to carry out this task.

The mechanical constraints in the optimisation involve the calculation of stress distribution. Four different types of stresses are used in the optimisation to check the design, as follows:

(i) Crushing stress describes the direct compressive stress on the teeth, which is defined as the mean compressive normal stress at the common edge between blade and disc. Bedding length, defined as the contact length between blade and disc, is the major factor affecting the stress.

(ii) Section stress, defined as the average stress along those minimum sections for each tooth of the blade and disc. This factor is related with the width of those sections and essentially provides a limit for the minimum width for both the blade and disc.

(iii) Unzipping can occur in the case of full blade release: the disc post on either side of the released blade are then subject to high tensile and bending stresses. The disc post must be able to withstand these stresses in order to avoid an 'unzipping effect' where all the blades are progressively released. The unzipping stress is defined as the stress at a point of 85% of length of the disc bottom section close to the empty slot of the disc, such as point A in Figure 2.1.

(iv) Peak stress: peak stresses can be found typically at the inner fillet radii of both the blade and the disc, where the stress concentration occurs. Since the stress concentration is a major contributor to the initiation of crack, the peak stresses provide a good indicator for stress concentration. If the fillet radii are too small and produce unacceptable peak stresses, some bedding width has to be sacrificed to make the radii bigger for both the blade and disc.

Apart from the above constraints, which are used to check the design, some others are normally used to check the optimised results. These include vibration limits, allowable neck stress, etc. From preliminary studies of blade number optimisation, these criteria are not deemed significant.[42]

As the firtree geometry is constant along the root centre-line, it is possible to think of the stresses as two-dimensional. However, the loading applied along the root centre-line direction is not uniform, so strictly speaking, the distribution of stresses will be three dimensional. Nonetheless, it is still possible to assume that each section behaves essentially as a two dimensional problem with different loadings applied on it. The difference of loading on each section is affected by the existence of a skew angle which will increase the peak stresses in the obtuse corners of the blade root and in the acute corners of the disc head. From previous root analysis research, it is feasible and convenient to use a factor to estimate the peak stresses at each notch of the blade and disc, and this factor takes different values for different teeth.

Also, it is known from previous work using photoelastic and finite element methods,[44-47] that the distribution of centrifugal load between the teeth can be non-uniform and the top tooth may take a larger proportion of the load. This feature allows the possibility of using different tooth sizes. The system implemented here also allows designers to explore the effect of varying the number of teeth, but this may cause difficulties when gradient-based methods are used for optimisation. This is because the number of teeth is not a continuous variable and hence that resulting objective function and constraints are not smooth with respect to this variable. Therefore, the number of teeth can only be varied in optimisations using genetic algorithm and direct search methods.

The whole process from the import of geometry, application of boundary conditions and loading, to results retrieval is implemented here as a SC03 plugin, which is a facility provided by the code to extend the capability of its core functionality. A command file is used by SC03 to carry out jobs ranging from importing geometry from an IGES file, applying boundary conditions and loads, to retrieving stress results. An unstructured mesh with six-nodal triangular elements is used throughout this work.

A one sector model and a three sector model are considered when estimating the mechanical constraints, the one sector model for the estimation of maximum notch stresses, crushing stresses, blade/disc neck mean stresses and the three sector model for the estimation of unzipping stresses. The cooling passage is modelled by using a reduced thickness value (13.50mm) for the area in the middle of the firtree root compared to a value used for elsewhere (32mm) to account for the effect of cooling passage. In general, finite element analysis is computationally expensive, thus a compromise between accuracy and computation cost should always be made to obtain acceptable results as quickly as possible when this is embedded in an optimisation run. This compromise is made by appropriate choice of mesh density.

The proper choice of global mesh control and local mesh control provides a good balance between computational cost and stress accuracy in high gradient areas such as tooth notch regions. Mesh control is achieved by specifying a number of mesh properties, which include node spacing and element distortion ratio. Default values of the analysis code are used for the global mesh properties. The size of an element for the notch region suggested by Dibsky *et al.*[45] should be less than 1% of the initial contact length. Meshes with different densities are shown in Figure 3.2. The notch stresses for different values of node spacing are shown in Figure 3.3 and 3.4 representing the fine and coarse mesh, respectively. The effect of mesh density can be shown in Figure 3.5, in which the result of a parameter study is shown. It can be seen from Figure 3.5 that an edge node spacing of 0.001 of the total length of the edge will be the largest allowable setting for a consistent notch peak stress.

Typical stress contour map for the one sector and three-sector model are given in Figure 3.6 and Figure 3.7, respectively.

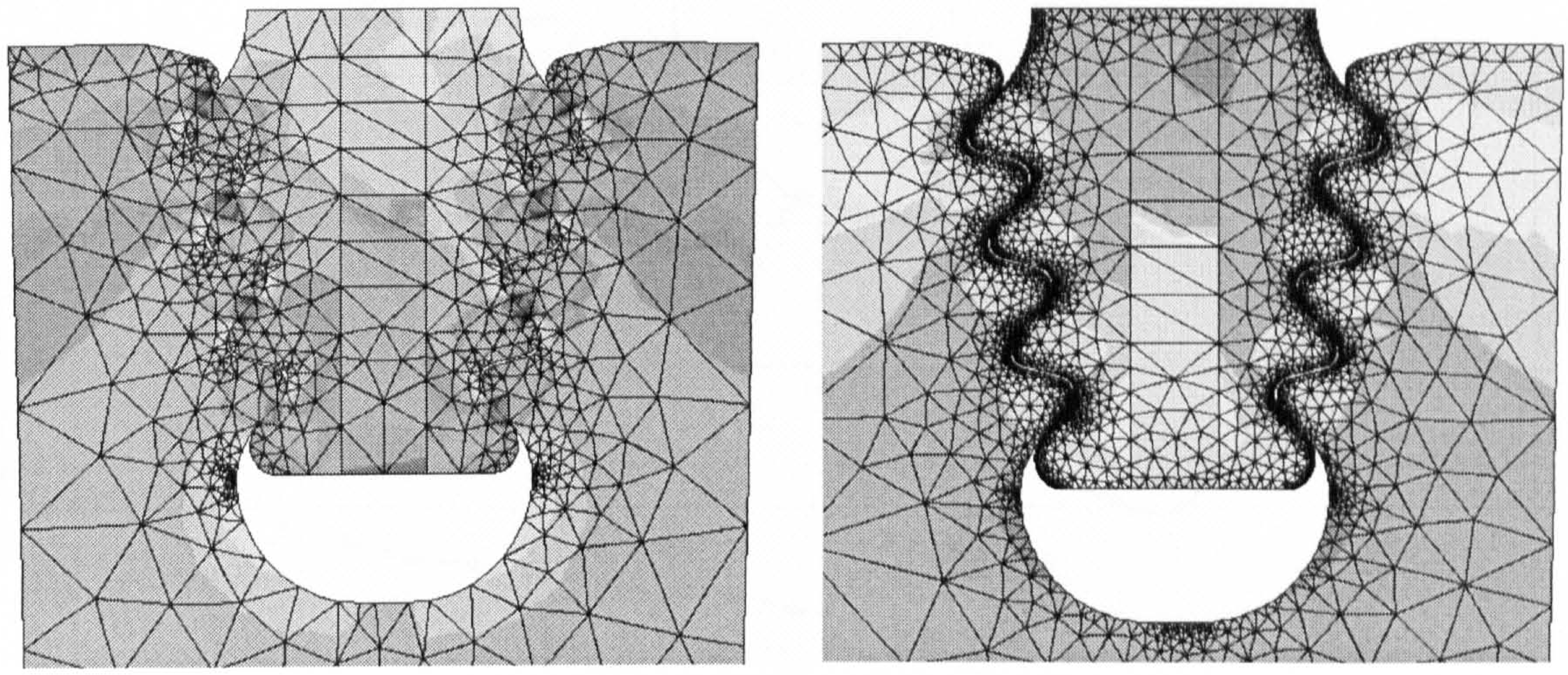


Figure 3.2: Comparison of finite element model with different mesh properties

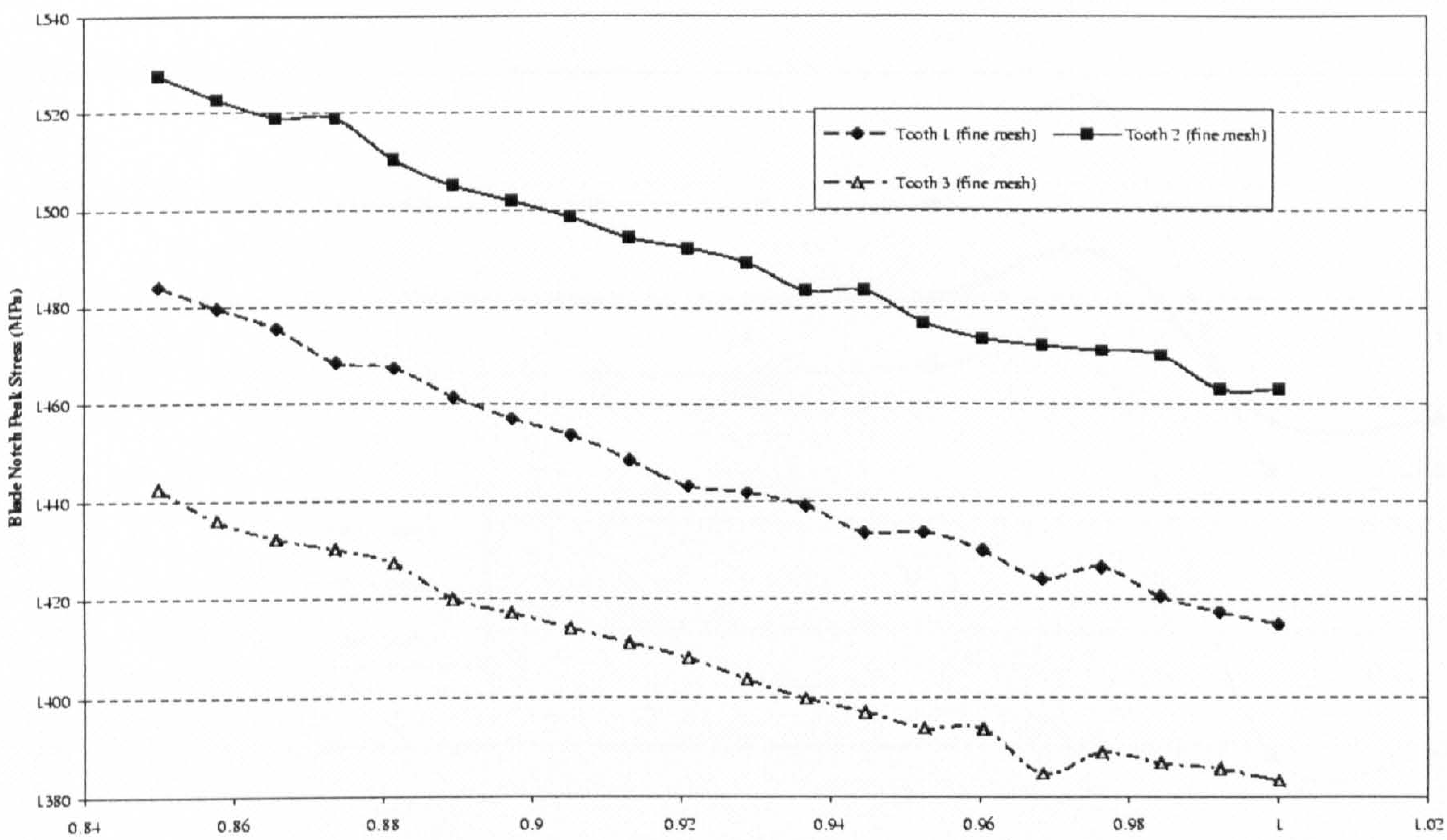


Figure 3.3: Notch stress variation against notch arc radius for fine mesh

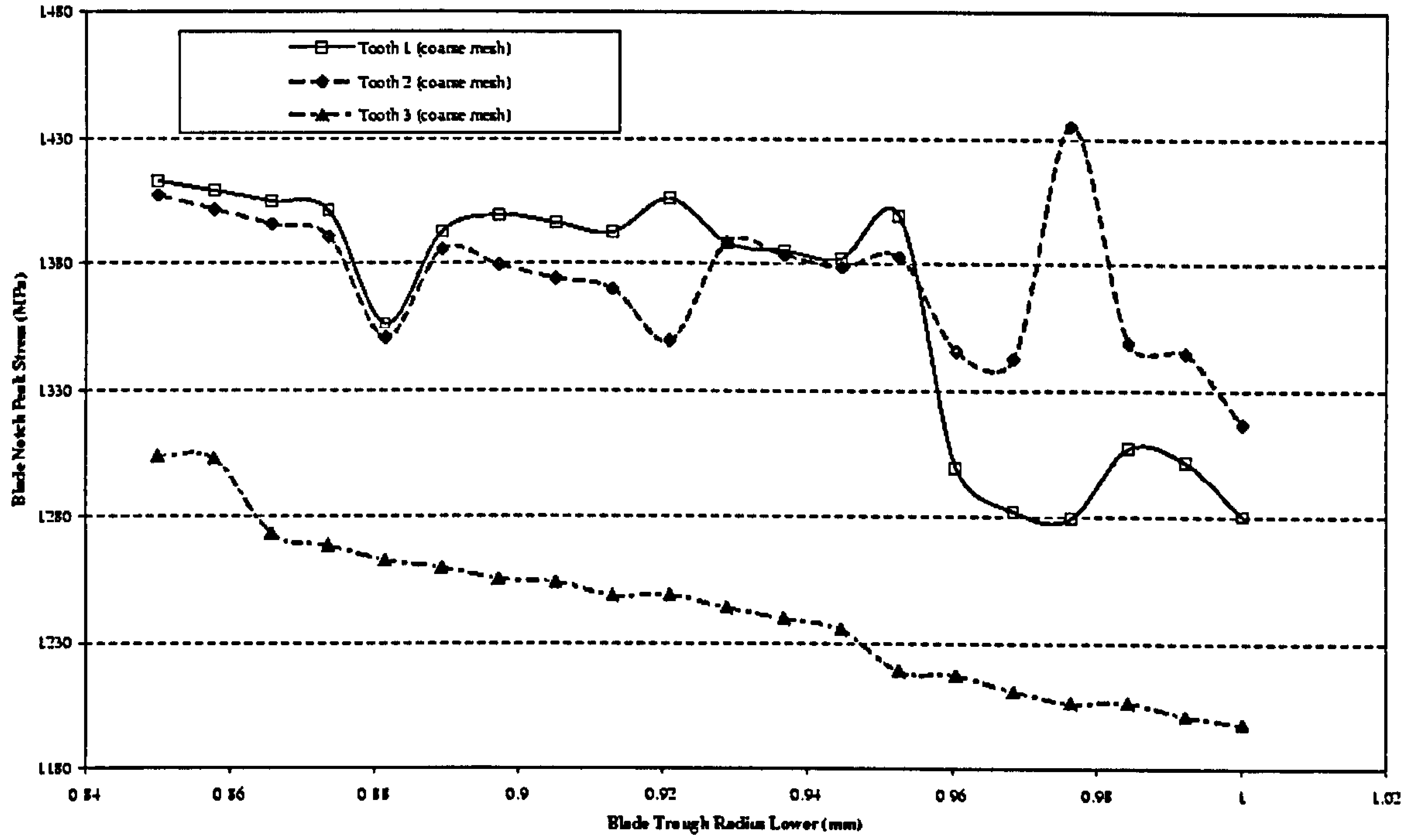


Figure 3.4: Notch stress variation against notch arc radius for coarse mesh

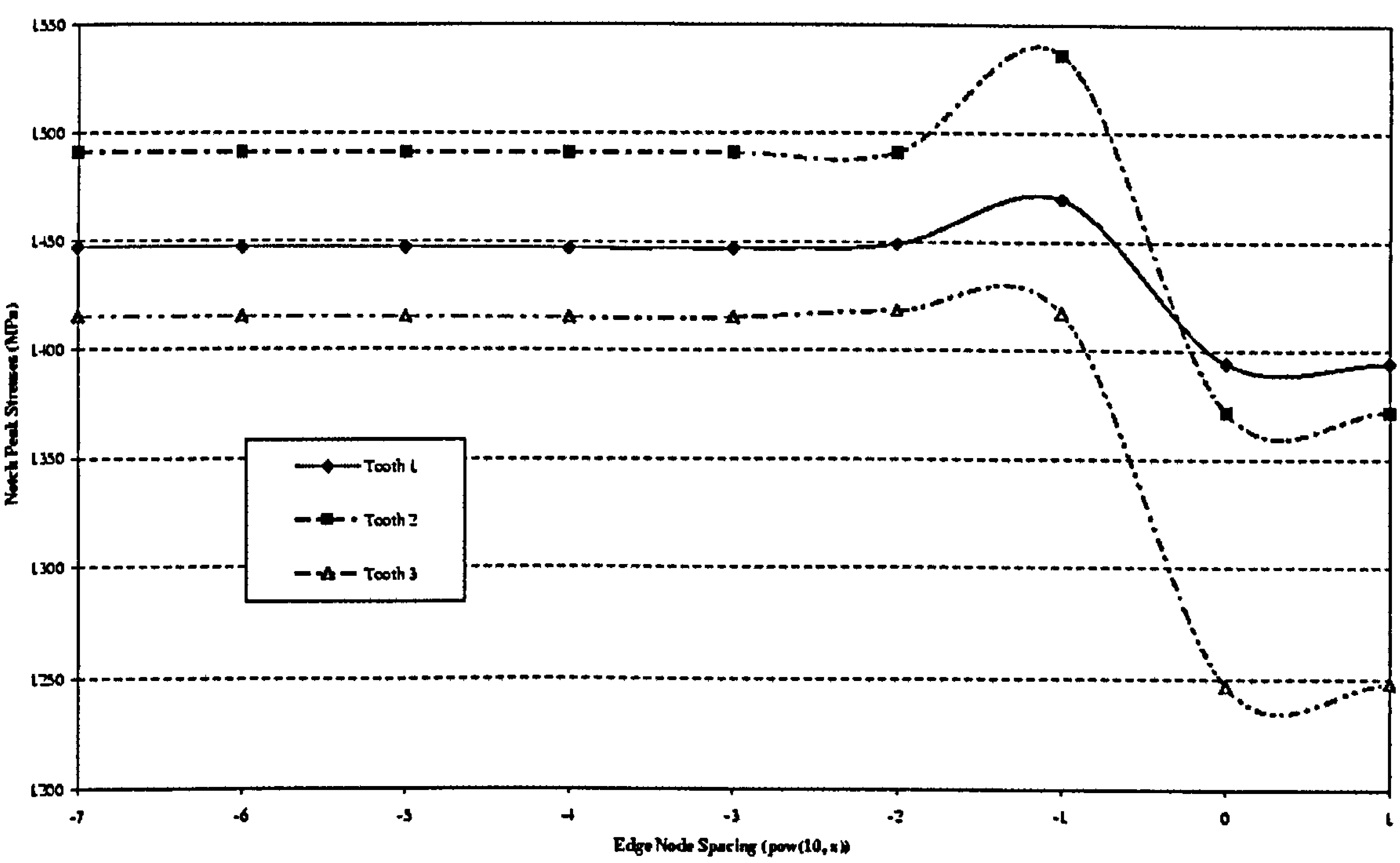


Figure 3.5: Notch stress variation against notch edge node spacing

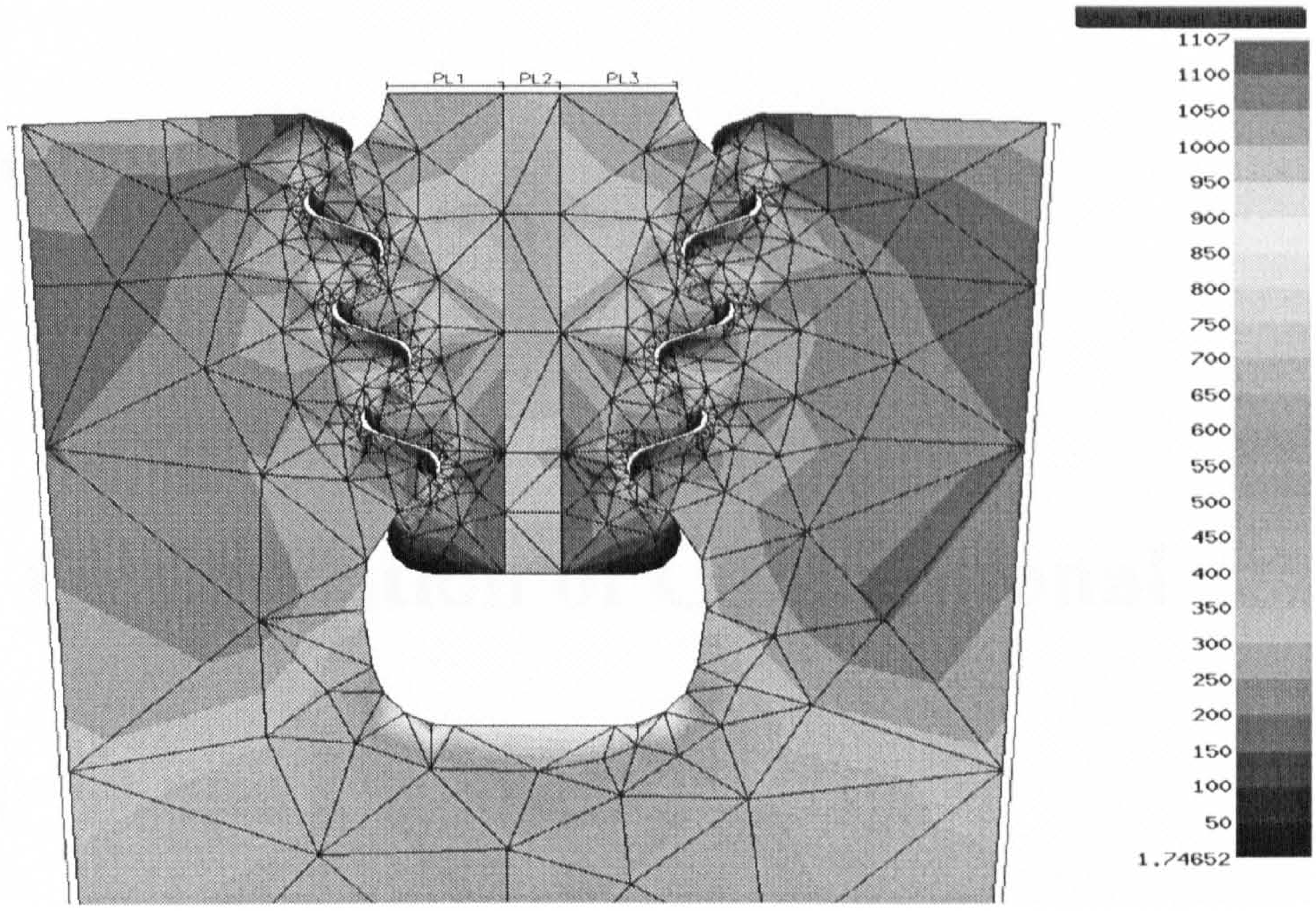


Figure 3.6: Typical stress field for single blade

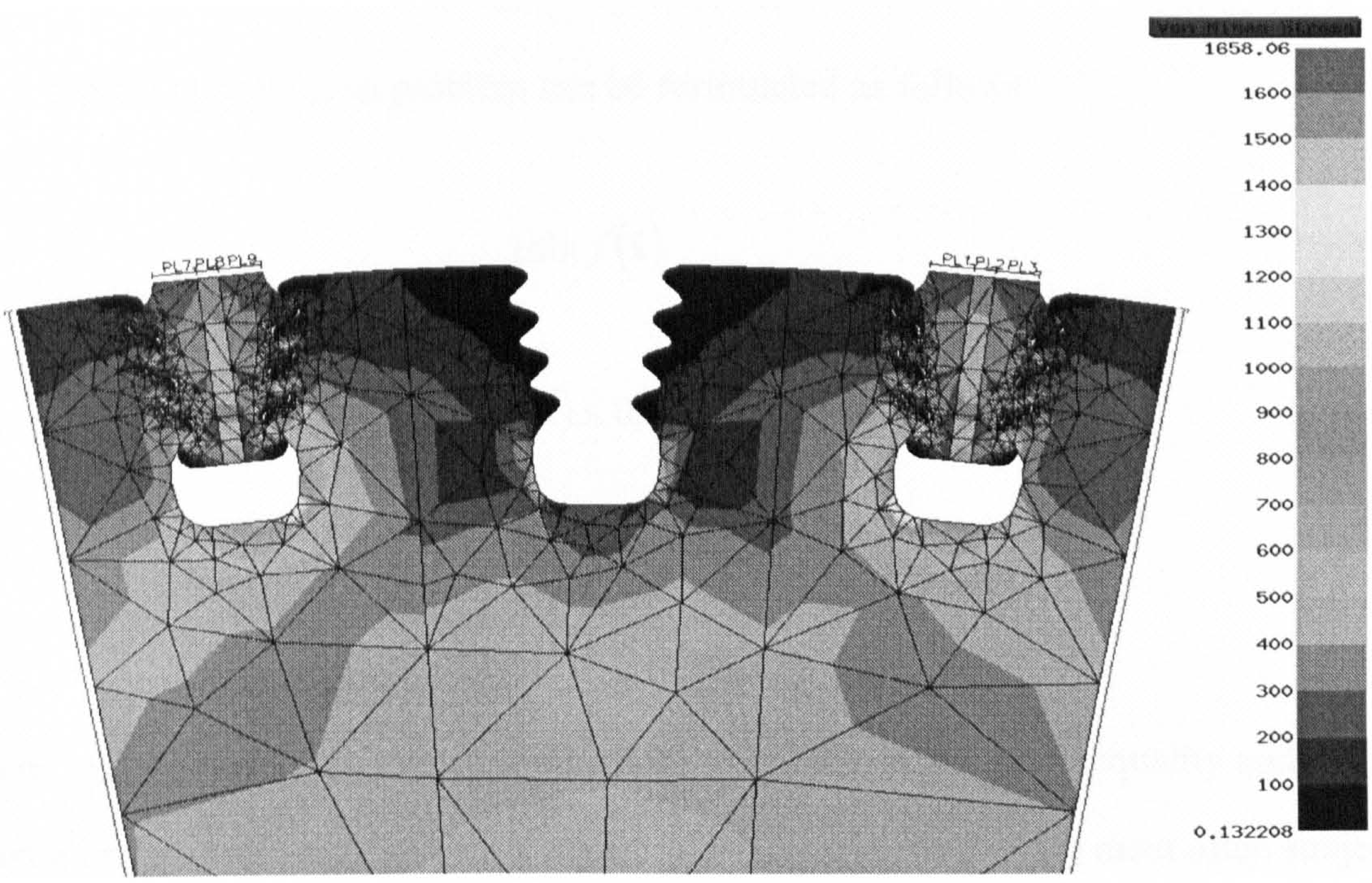


Figure 3.7: Typical stress field for three blade model with the middle blade out

Chapter 4

Direct Optimisation of Conventional Firtree Shape

4.1. Introduction

A typical optimisation problem can be formulated as follows:

$$\min f(\vec{\mathbf{x}})$$

$$\text{subject to } g_i(\vec{\mathbf{x}}) \le 0, (i = 1,\ldots,p) \text{ and/or} \tag{4.1.1}$$

$$h_j(\vec{\mathbf{x}}) = 0, (j = 1,\ldots,q)$$

where $f(\vec{\mathbf{x}})$ represent the objective function, $g_i(\vec{\mathbf{x}})$ and $h_j(\vec{\mathbf{x}})$ are inequality and equality constraints, respectively. $\vec{\mathbf{x}}$ is the vector of design variables that are most often subject to bound constraints $\vec{x}_k^l < \vec{x}_k < \vec{x}_k^u, (k = 1,\ldots,n)$. The task of optimisation is to determine the set of parameters $\vec{\mathbf{x}}$ given by $\vec{\mathbf{x}} = \text{argmin} f$ subject to the constraints above.

There are a large number of different optimisation methods available today, either classic gradient-based search procedures or modern stochastic evolutionary methods. The choice of methods is not a simple issue, as the performance of any particular method is strongly affected by the characteristics of the problem, such as whether there are constraints or not, whether the constraints are linear or not, whether the objective function is smooth or not, whether multiple optima exist or not, etc.[10]. In a typical engineering optimisation problem, the objective functions or constraints are often evaluated by complex simulation codes. A number of challenges associated with the optimisation of complex engineering systems have been identified by Eldred[51], which include interfacing optimisation methods with simulation codes, the high cost required for a single evaluation, unavailability of analytic derivatives, presence of noise in the objective function or constraints, and mixes of discrete and continuous variables, etc.

Optimisation techniques can be broadly classified into two categories; one is the traditional search methods which utilize derivative information and work in a downhill fashion. Methods falling into this category include steepest-descent methods, Newton and Quasi-Newton methods, and Sequential quadratic programming[10,52,53]. Although these methods are known to be efficient in locating a better solution, they are more likely to stop at local optima and are also sensitive to the noise present in the outputs of many engineering problems. Multi-start approach, often implemented in parallel, can be adopted to avoid the pitfalls of converging to a local optimum[54-57]. The other type is the evolutionary and stochastic computation methods that may include Genetic Algorithms[16], Evolutionary Strategy[18], Simulated Annealing[17] which do not proceed directly downhill.

Developed and matured in the last two decades, evolutionary algorithms prove to be suitable for multi-modal problems, particularly with discrete search spaces. Various types of evolutionary algorithms exist, among which, Genetic Algorithms are perhaps most extensively studied in the literature, also, there is rigorous analysis for the behaviour analysis of these algorithms; for example, the schemata theory for the binary coding genetic algorithms.[16] Genetic algorithms are designed to mimic evolutionary selection in nature. Each individual is represented by a binary string, which corresponds to the set of design variables. A population of candidate designs is formed at the beginning and evolved using genetic operators such as crossover and mutation. When individuals are selected for generating offspring, parts of their binary string are exchanged by crossover operation. There is also a probability that some bits are modified according a mutation operator, which is essential for maintaining diversity in the population, therefore avoiding premature convergence. Genetic algorithms have been shown to be effective in various engineering optimisation problems.[58,59]

There is also growing interests in recent years in hybridising different types of search methods to formulate various strategies to improve the performance of search procedures in terms of both efficiency and robustness of finding global optimum of single techniques.[60-62] A two-stage procedure, a typical genetic algorithm followed by gradient search on the promising individuals is presented in this chapter and applied to the firtree problem.

This chapter is organised as follows: interfaces between modelling, analysis and optimisation package is first described in the next section 4.2. Section 4.3 and 4.4 define the geometric and mechanical constraints used in the problem. In section 4.5, results of parameter study are presented. Section 4.6 presents the main results of direct optimisation of the firtree problem using two-stage search procedures.

4.2. Interface between optimiser, modelling and analysis

The overall architecture of the design optimisation used here is illustrated in Figure 4.1. In this structure, ICAD[7] is used to generate the model definition based on rules which are coded in a number of lisp files. The model is defined in a descriptive form using the ICAD design language (IDL) and is used to produce analysis-related data as well as the geometry. The geometry is then passed to the analysis code along with geometry dependent properties to evaluate the design performance. The design exploration system OPTIONS[11] is used to find the best solution for a particular set of design variables. Parallel processing can be achieved by using facilities provided by OPTIONS, in this case the number of ICAD licenses is a limiting factor and the jobs wait until an ICAD license is available.

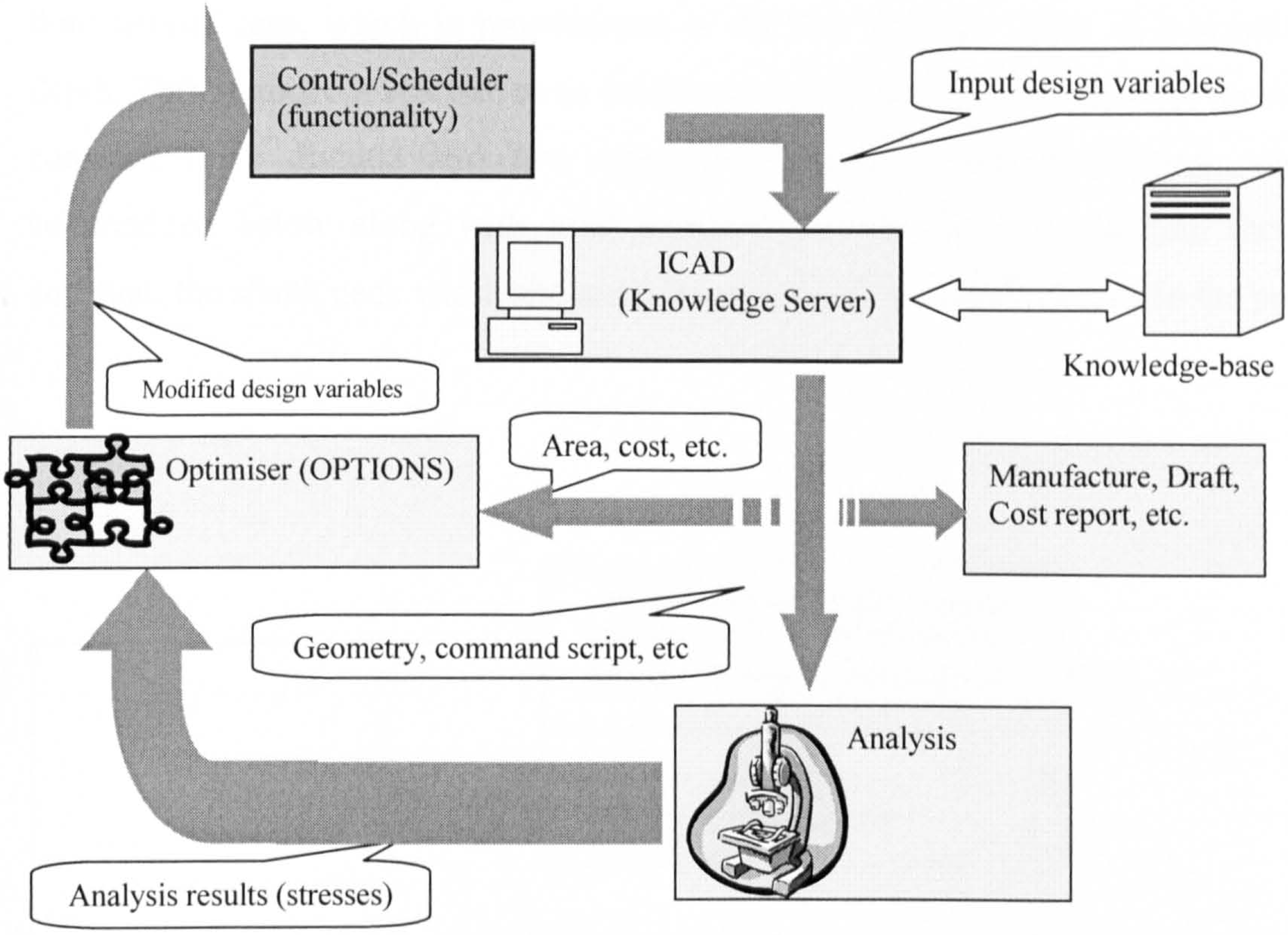


Figure 4.1: Overall System Architecture of ICAD-Based Design Optimisation

The OPTIONS package provides designers with a flexible structure for incorporating problem specific code as well as with more than forty optimisation algorithms. The critical parameters to be optimised, known as the design variables, are stored in a design database, which also includes the objective, constraints and limits. The design variables are transferred to ICAD by means of a property list file which contains a series of pairs with alternating names and values. This file is updated during the process of optimisation and reflects the current configuration. The geometry file produced by ICAD is then passed to the FE code SC03 which is executed by a command file. The analysis results are written out to another file, which is read in by the optimisation code. The design variables are then modified according to the optimisation strategy in use until convergence or a specified number of loops has been executed. The program structure is illustrated in Figure 4.2.

The design of the firtree structure involves many geometric parameters with complex relations, and evaluation of the performance of the firtrees cannot be confidently carried out without using finite element codes. The initial objective in the firtree problem is to minimise the area outside of the last continuous radius of the turbine disc for the two-dimensional case, which is proportional to the rim load by virtue of a constant axial depth. This quantity is referred to as the firtree frontal area in the following sections. The constraints are divided into two categories, geometric and mechanical, which are summarized below along with brief explanations. In order to maintain the loading constant, the shank neck width and axial length of root are kept constant in the problem.

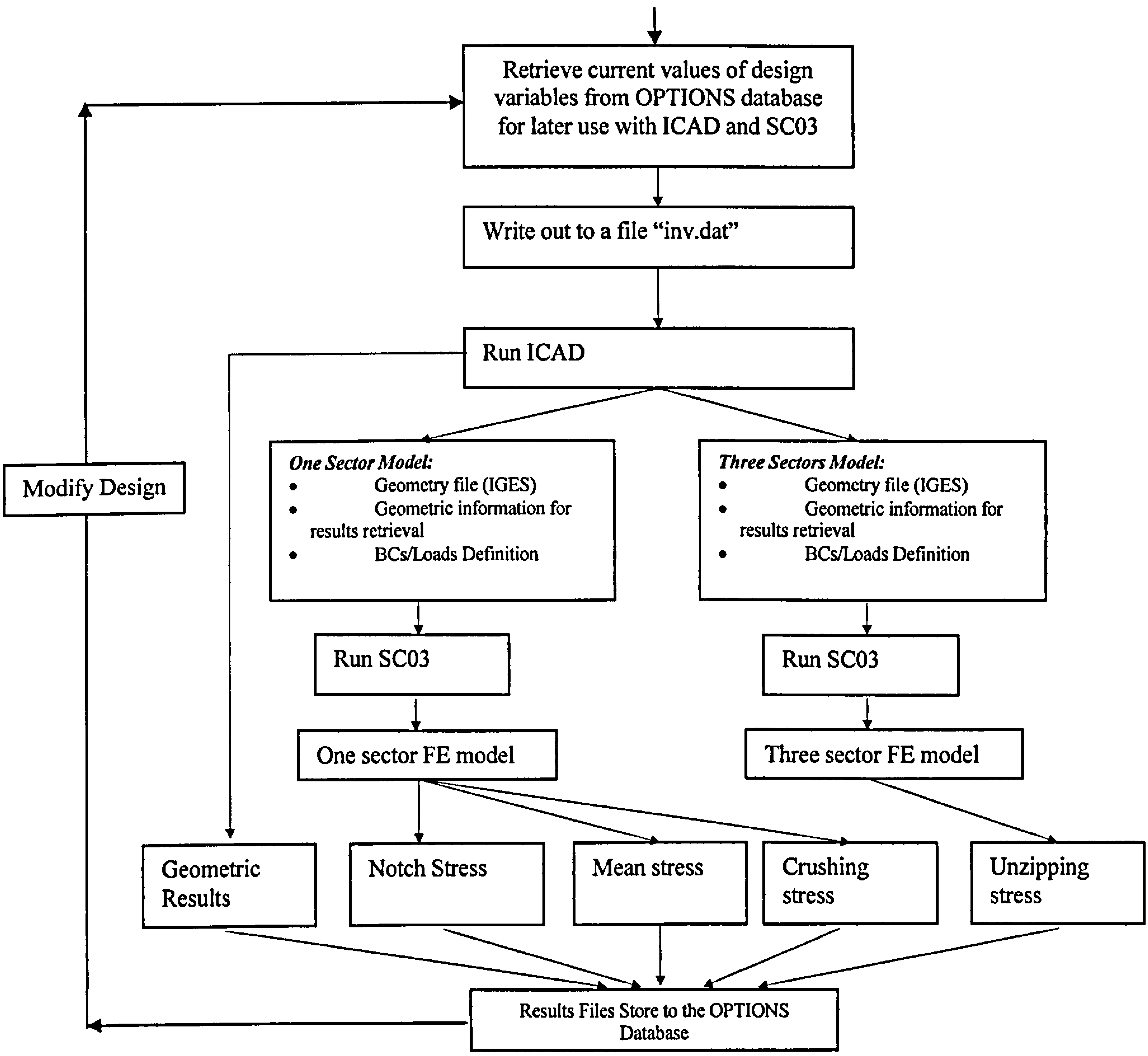


Figure 4.2: Detailed Optimisation Program Structure

4.3. Geometric constraints

The geometrical constraints imposed in the firtree design problem are mainly derived from industrial experience[42]. The meaning of some of these symbols used in this section can be found in Figure 2.1 and Figure 2.2 in Chapter 2.

1. $R_1 / R_2 > R_{12\min}$ to have a failsafe design by ensuring that R_2 will fail before R_1 to avoid disc burst, $R_{12\min}$ represents the lower limit for the ratio;
2. $H / D > HD_{\min}$ to discourage cracks from the bucket groove travelling into the disc diaphragm, $HD_{\min}$ represents the lower limit for the ratio;
3. $R_1 / F_{trough} > R1F_{\min}$ to try and make the firtree fail before the bucket groove, $R1F_{\min}$ indicates the lower limit of the ratio;
4. $DPF_{\max} > DP / F_{trough} > DPF_{\min}$ to balance serration and bedding stresses within 30° flank angle, $DPF_{\min}$ and $DPF_{\max}$ represent the lower and upper limits of the ratio;
5. $L / C_a > LC_{\min}$ to prevent stress concentration in the aerofoil root spreading into the shank blend region, where L and C_a represent the axial length of firtree root and , $LC_{\min}$ is the lower limit of the ratio;
6. Root skew angle should meet the maximum requirement;
7. Blade and disc serration pitch should be no less than the specified value;
8. Blade bottom neck width should be no less than serration width;
9. Minimum wall thickness requirement at bottom blade notch requirement on cooled blades;
10. Bucket groove area should be greater than cooling passage area in the blade root.

4.4. Mechanical constraints

An additional set of stress constraints also based on past experience[42] are imposed, these mechanical constraints are listed below:

11. Blade and disc peak stresses at notch areas less than the material limits;
12. Section stresses of the blade and disc sections at each tooth less than the prescribed upper limits;
13. Crushing stresses must be less than the allowable values for a given root form;
14. One blade out event must be considered: σ unzipping at worst operating condition less than minimum UTS for coarse lobe root, σ unzipping is calculated from σ bending at extreme fibre and σ direct;

These constraints are listed in Table 4.1 along with the normalized values of the constraints for a base design. The base design, derived from an existing product model, is analysed first. The geometry of the base design is defined using straight lines and circular arcs and values of those parameters are given in Table 2.1. The analysis results revealed that several geometric and mechanical constraints are violated. These include geometric constraints 4, 8, 9 and 10, and disc notch stress constraints. The geometric constraint 4 indicates that a smaller tooth pitch and/or a larger disc trough radius are required to satisfy this constraint. Also, decrease in tooth pitch would push the constraints 9 and 10 towards feasible region as a smaller tooth pitch will result in a bigger blade bottom neck and wall thickness of bottom blade notch. However, connections between these three violated geometric constraints and other design variables cannot be easily identified.

The reason for the violation of mechanical constraints arises from using a criterion based on the scaled Ultimate Tensile Stress (UTS) data due to the lack of fatigue data. This is a conservative option compared to a criterion based on the fatigue data. These constraints will then become the limiting factors for the optimum design found by the optimiser. Due to the unavailability of fatigue data, a scalar factor has been introduced to relax the stress constraints and to make the current design feasible in terms of stress constraints. Table 4.1 shows the resulting normalized constraint values for the base design when using this factor.

Table 4.1 Normalized Constraint vector for the base design

Name of constraint	Numeric values		
	Lower Bound	Value	Upper Bound
Ratio of R1 to R2 [R1/R2]	-1.0	-0.8642	-
Ratio of H to D [H/D]	-1.0	-0.8955	-
Ratio of R1 to disc trough [R1/Ftrough]	-1.0	-0.5268	-
Maximum ratio of tooth pitch to disc trough [DP/Ftrough(max)]	-	1.4779	1.0
Minimum ratio of tooth pitch to disc trough [DP/Ftrough(min)]	-1.0	-0.5467	-
Ratio of axial length to blade axial chord [LCA]	-1.0	-0.4961	-
Root Stagger Angle [RSA]	-	0.7499	1.0
Ratio of Blade/Disc serration pitch [PMIN]	-1.0	-0.4540	-
Ration of blade bottom neck width to tooth pitch [BNP]	-1.0	-1.1474	-
Minimum wall thickness of bottom blade notch [BNMIN]	-1.0	-1.3038	-
Ratio of bucket groove region area to cooling passage area [AR]	-1.0	-1.0900	
Maximum blade notch stress [NBL(R)(2)]	-1.0	0.9270	1.0
Maximum disc notch stress [NDL(R)(3)]	-1.0	0.9948	1.0
Maximum blade section stress [SB(1)]	-1.0	0.6931	1.0
Maximum disc section stress [SD(4)]	-1.0	0.5623	1.0
Maximum crushing stress [CS(1)]	-1.0	0.6514	1.0
Maximum unzipping stress [UZP(1)]	-1.0	0.3688	1.0

Notes: (1) For the purpose of compactness, only the maximum stresses are shown in the table. (2) The numbers in brackets indicate the no. of the tooth or the section where the maximum stress occurs.

Note that the constraints on disc notch stresses are among the closest to violation at this stage. The normalization adopted here is described as follows:

constraints with upper bounds only: $y_{norm} = \begin{cases} y/u, (u \neq 0) \\ y(u=0) \end{cases}$;

constraints with lower bounds only: $y_{norm} = \begin{cases} -l/y, (y \neq 0, l > 0) \\ -y/l, (l < 0 \quad or \quad y = 0) \\ y(l=0) \end{cases}$;

constraints with both upper and lower bounds: $y_{norm} = 2\left(\frac{y-l}{u-l}\right) - 1$.

Normally, these formula will make it possible for all normalized constraints to have consistent behaviour when the design moves from an infeasible region towards a feasible region, and to have the values of -1 or $+1$ at the boundary of the constraints.

4.5. Results of Parameter study

Following the set up of the system, a series of parameter studies were carried out to establish appropriate parameter values and to gain experience on the behaviour of the response quantities. These were performed by varying one variable at a time while holding all others constant. It was also necessary to find out the appropriate mesh density parameters required to capture the notch stresses. Therefore, the local and global effects of mesh density were studied. These were achieved by varying the mesh density set in the ICAD model and then running an analysis. These studies resulted in the use of 0.001 of the edge length for both global and local edge node spacing which explicitly controls the node spacing along the geometry edges. Parameter studies on the geometric parameters revealed the effect of different geometric features on the stress distribution within the structure, which are summarized in Table 4.2. It can be seen from these results, as shown in Figure 3.3, that the notch stress on the second tooth, and not the bottom tooth takes the largest value, as already implied from previous work:[23] this makes it desirable to model each tooth using different values of tooth profile parameters.

It can be seen from these results of the parameter study that the effects of some geometric features on one type of stresses are greater than on other types of stresses, and some geometric features have apparent effects on more stresses than others. For example, only a small number of geometric features have apparent effects on the section stresses, as can be seen from the section stress column in the Table 4.2; the tooth pitch has significant effects on almost all types of stresses and bucket groove radius hardly have any significant effect on the stresses. These observations can be used to either reduce the number of design variables or group those geometric parameters having biggest impact together when investigating some particular stresses.

4.6. Optimisation Results

Two different optimisation problems are tackled here using population based genetic algorithms and gradient-based methods. One was to minimize the firtree frontal area as defined at the beginning of this section, which is related to the weight, subject to the geometric and mechanical constraints. The other was to find the optimum tooth profile to minimize the maximum notch stress after the area is minimised. The design variables (Table 4.3) and constraints (Table 4.4) used in the second problem are a subset of those defined in the first problem which has 14 design variables and up to 53 constraints for a three-tooth design. Since some of the parameters are fixed in the second optimisation, the changes are limited to the notch profile therefore deemed as local changes. The reduction in peak stresses potentially implies the increase of fatigue life as the stress concentration factor is minimised. During optimisation, each evaluation takes about 5-6 minutes to finish on a LINUX machine with dual 800Mhz CPU and 512 Mbytes memory, and most of this time is engaged in finite element analysis. This means that it takes about 80 hours to finish a 10 generation GA search using a population size of 100. Note that for some specific sets of parameters, there is no feasible geometry that can be constructed, and when this occurs SC03 is simply signalled to cancel the analysis.

Table 4.2 Results from parameter study on base design

Geometry features	Unzip stress	Notch stress		Section stress		Crushing stress
		blade	disc	blade	disc	
Skew angle	X	+	+	+	X	1+,restX
Shank neck width	-	X	1+;restX	+	X	1,4-;restX
Blade shoulder width	+	-	2,3+;rest-	1+;rest-	+	+
Bottom tooth crest radius	X	2+, rest-	-	X	X	4+;restX
Cooling passage width	X	1,2+;rest-	1-;2,3+	+	X	1,2+;3,4-
Bucket groove lower radius	X	X	X	X	X	X
Bucket groove upper radius	X	X	X	X	X	X
Tooth pitch	+	+	+	X	+	-
Blade crest radius	1-;rest+	-	+	X	X	+
Blade trough radius	-	+	-	X	X	1,2-;3,4+
Disc crest radius	1-;rest+	-	-	+	-	+
Disc trough radius	-	X	-	X	X	4-;rest+

Note: + indicates increase, - indicates decrease, X means no significant effect when variable increases, number indicates the tooth number in top-down order

Because of the large number of design variables, optimisation traces may not be simply plotted on contour maps of any two variables, as these maps are necessarily produced while holding all the other variables constant. Furthermore, if only a small number of quantities are chosen as design variables, there may be no feasible designs at all. For an infeasible starting design, it is easier for the optimiser to find a feasible region if a large number of quantities are left as design variables and broad exploratory search methods are used. Therefore a typical genetic algorithm (GA) in the OPTIONS as discussed by Goldberg,[16] with extension to incorporate a version of clustering algorithm[63] is first employed, which has been applied with some success to multi-peak problems.[64] The main objective here is to give a fairly even coverage on the search space, and then gradient based search methods are applied on promising individuals. The use of this strategy is based on the idea that as the GA proceeds, the population tend to saturate with designs close to all the likely optima including sub-optimal and globally optimum designs, which form different clusters, while gradient based methods are efficient at locating the exact position of individual optimum, i.e. the GA results give good starting points for gradient search methods. Although other methods like Design of Experiments could also be used in identifying useful starting points for gradient searches, the quality of these starting points bounds to be poorer than the ones returned from many generations of GA runs. One of the factors which affect the choice between these two methods is the available computational budget and cost of one function evaluation.

A contour map for two design variables has been generated, as shown in Figure 4.3. Note that the optimum shown on such figures may not be the global optima as the other design variables are being varied by the optimiser. Infeasible geometries and analysis failures are also illustrated in Figure 4.3. It is noted that identification of analysis failure is useful for identifying any problems with the analysis code, but this is not a concern for optimisation as long as appropriate measures are employed to avoid misleading the optimiser. In the implementation of this problem, the value of the objective function is directly defined using different values according to the situations while constraints are simply regarded as being violated for such points. This is done by setting a maximum time target for the analysis code, any analysis job that lasts longer than this target value will be deemed as failure, and an appropriate value will be set for the objective function in this case. This is important especially when approximations such as response surfaces are introduced to improve run speeds. A gradient based search is illustrated in Figure 4.4. It can be seen that better starting points do not always converge to better results, depending on the location in the design space. This justifies the use of more promising individuals in the final population of the GA results instead of just the best one as starting points for gradient search.

Table 4.3 Definition of optimisation problems

Variables	number of variables	
	6	14
	Tooth profile parameters	
root wedge angle (degree)	20-40	20-40
Tooth pitch (mm)	2.0-4.0	2.0-4.0
Blade crest radius (mm)	0.2-1.0	0.2-1.0
Blade trough radius (mm)	0.2-1.0	0.2-1.0
disc crest radius (mm)	0.2-1.0	0.2-1.0
disc trough radius (mm)	0.2-1.0	0.2-1.0
	Firtree root /disc head parameters	
Skew angle (degree)	[15]	10-20
Axial length of root (mm)	[20]	15-25
shank neck width (mm)	[6.7615]	6.5-7.5
firtree shoulder width (mm)	[9.8943]	8-12
bottom tooth crest radius (mm)	[1.0668]	0.8-1.2
cooling passage width (mm)	[1.3455]	1.2-1.5
bucket groove lower radius (mm)	[3.5]	3.0-4.0
bucket groove upper radius (mm)	[2.2]	1.5-2.5

Note: [] indicates values from a base design are used when not varied in optimisation.

Several search methods from OPTIONS have been applied to the problem after the GA search, these include the Hooke and Jeeves method[65] plus various other methods discussed in Schwefel's book.[52] The first method is very fast when the number of design variables is small, as shown in Figure 4.5, in which only 6 variables are chosen as design variables, while the full-scale problem contains 14 design variables. Although the complexity of this problem is only modest, the computational cost in terms of thousands of evaluations required for some search techniques is still an obstacle for a detailed search. It can be seen from the contour maps that the objective function is rather smooth, and this may justify the use of approximation techniques alongside the accurate finite element model. This will be the focus of chapter 6.

A 20% reduction in the objective function was achieved for the firtree frontal area using the above methods while all the geometric and mechanical constraints were satisfied. Table 4.5 gives the values of design variables for both the base design and several resulting designs obtained using various methods. It can be seen from the first two columns of Table 4.5 that optimisation using genetic algorithm has produced a feasible design satisfying the violated constraints of the base design. However, there is little improvement in the objective function, i.e. the firtree frontal area of the GA result is almost the same as that of the base design. Further searches starting from several best GA results produced more promising results in terms of both the objective function improvements and constraints satisfaction, as revealed by the last two columns of Table 4.5. Comparing the final results and base design, several apparent changes can be observed which include a decrease in tooth pitch, an increase in cooling passage width, and a decrease in disc trough radius, disc crest radius, blade crest radius. These primary changes to geometry can be clearly seen in Figure 4.6. In addition, the root wedge angle is slightly increased.

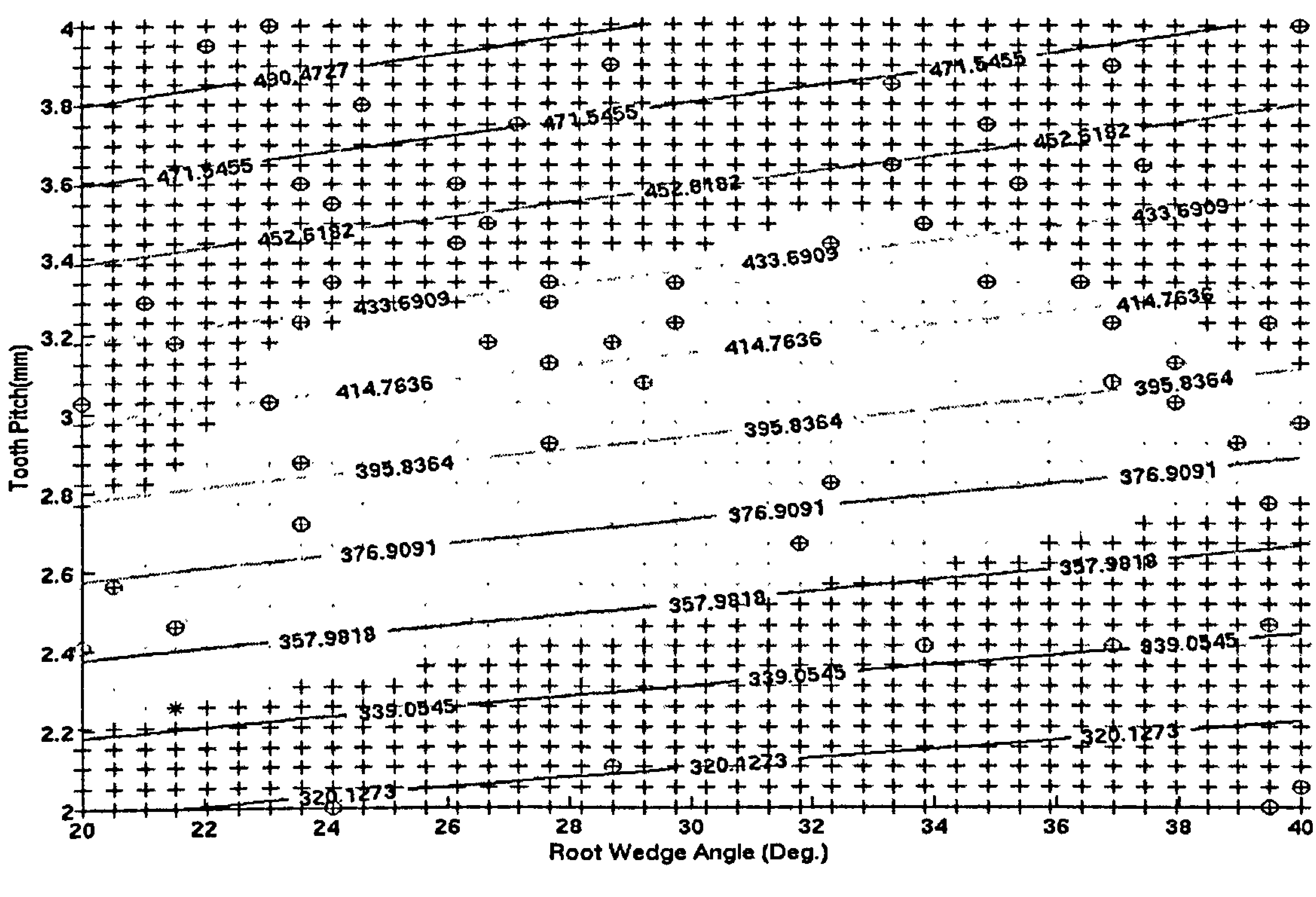


Figure 4.3: Contour map of frontal area for root-wedge-angle and tooth-pitch
⊕analysis failure + infeasible points * minimum as shown

Following the search on the full-scale problem of minimising the firtree frontal area, the second optimisation problem with the objective function set to minimize the maximum notch stress was carried out. The result is shown in Table 4.6 and Figure 4.7. The six tooth profile parameters were chosen as design variables instead of the complete set of parameters used in the full problem. It can be seen that although an improvement of 25% reduction in the maximum notch stress can be achieved, the firtree area is increased by approximately 11%. Also note that the root wedge angle has dropped significantly while the pitch and all the radii have risen. Clearly, the choice of objective function has significant impact on the final design, and multi-objective optimisation techniques may be useful when conflicting objectives are involved.

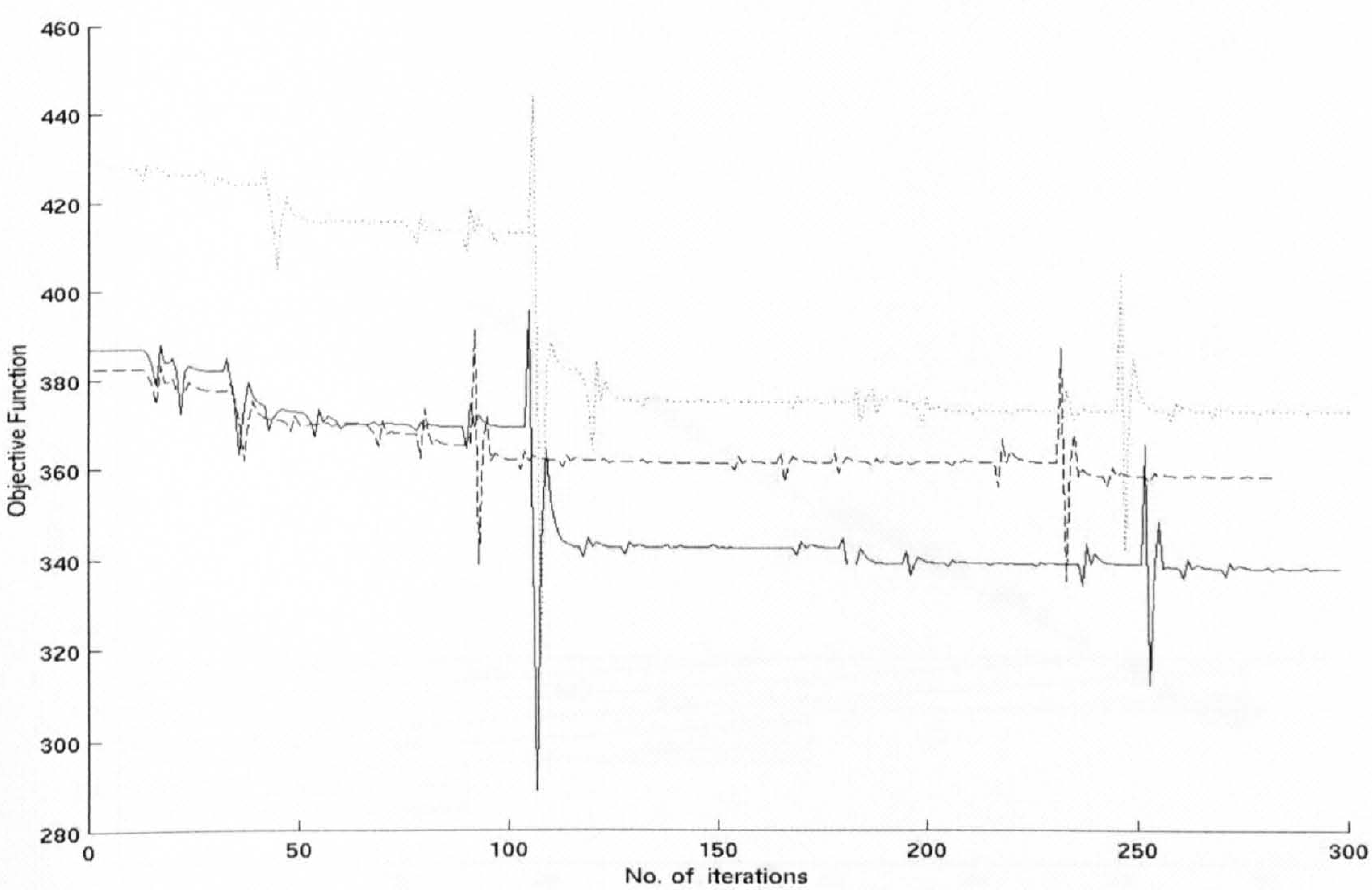


Figure 4.4: Schwefel Library Method: Repeated Lagrangian Interpolation (LAGR) on GA results

Table 4.4 Constraints and objective applied in the optimisation problem

Number	Constraints	Number of Variables	
		6	14
1	R1/R2	X	X
2	H/D	X	X
3	DP/F	X	X
4-5	min<DP<max	X	X
6	Lca	X	X
7	RSA	X	X
8	minimum serration pitch	X	X
9	bottom neck width > pitch	X	X
10	minimum wall thickness	X	X
11	Bucket groove area> cooling passage area	X	X
12-23	Notch stresses		X
24-35	Section stresses	X	X
36-43	Crushing stresses	X	X
44-45	Bucket groove stresses	X	X
46-53	Unzipping stresses	X	X
Objective		Notch stress	Firtree frontal area

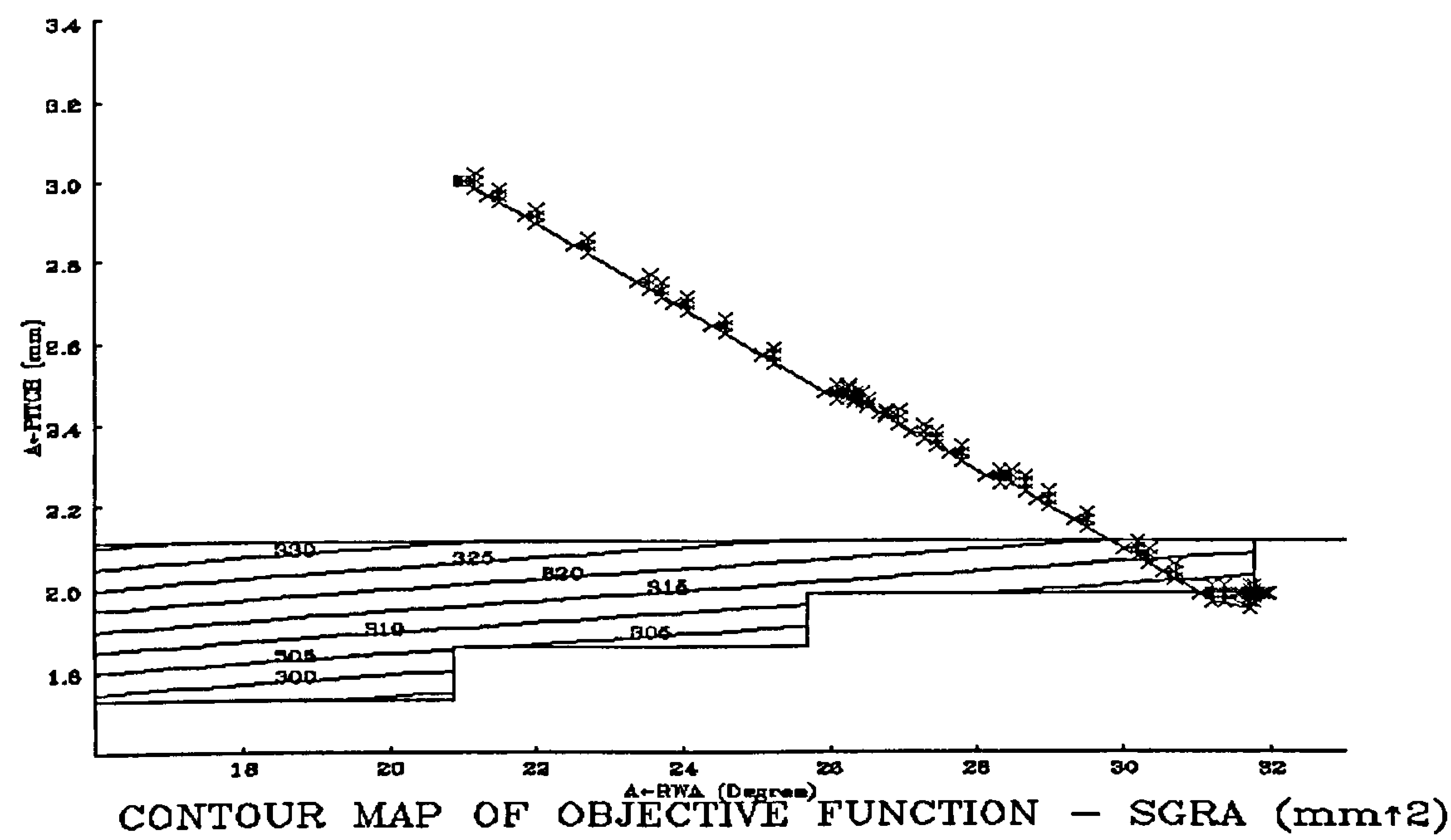


Figure 4.5: Hooke and Jeeves search results based on GA search results

Table 4.5 Search results using various classic methods

Design Variables	Starting point	Best results using GA method	Lagrange interval search after GA	Best results obtained using LAGR after GA
skew(degree)	15.0000	17.8272	17.8272	17.9054
alor(mm)	25.4000	28.3694	28.3694	28.2131
snw (mm)	6.7615	6.6322	7.3280	7.3311
fsw (mm)	9.8943	11.2618	9.6405	9.5210
btcr (mm)	1.0668	1.02596	0.8448	0.8182
cpw (mm)	1.3455	1.42896	1.4853	1.4859
bglrl (mm)	3.5000	3.16856	3.0642	3.0596
bgur2 (mm)	2.2000	1.6395	2.3093	2.3099
a_rwa (degree)	31.8428	31.6844	29.4738	33.5598
a_pitch (mm)	3.3038	3.3036	2.9094	2.5615
a_rcrest (mm)	0.5461	0.5460	0.9385	0.5492
a_fcrest (mm)	0.5969	0.5968	0.5511	0.5468
a_rtrough (mm)	0.4699	0.4698	0.3046	0.3698
a_ftrough (mm)	0.4191	0.4190	0.3925	0.3016
firtree frontal area mm^2	423.07	425.44	368.56	337.64
No. of constraints violations	5	0	0	0

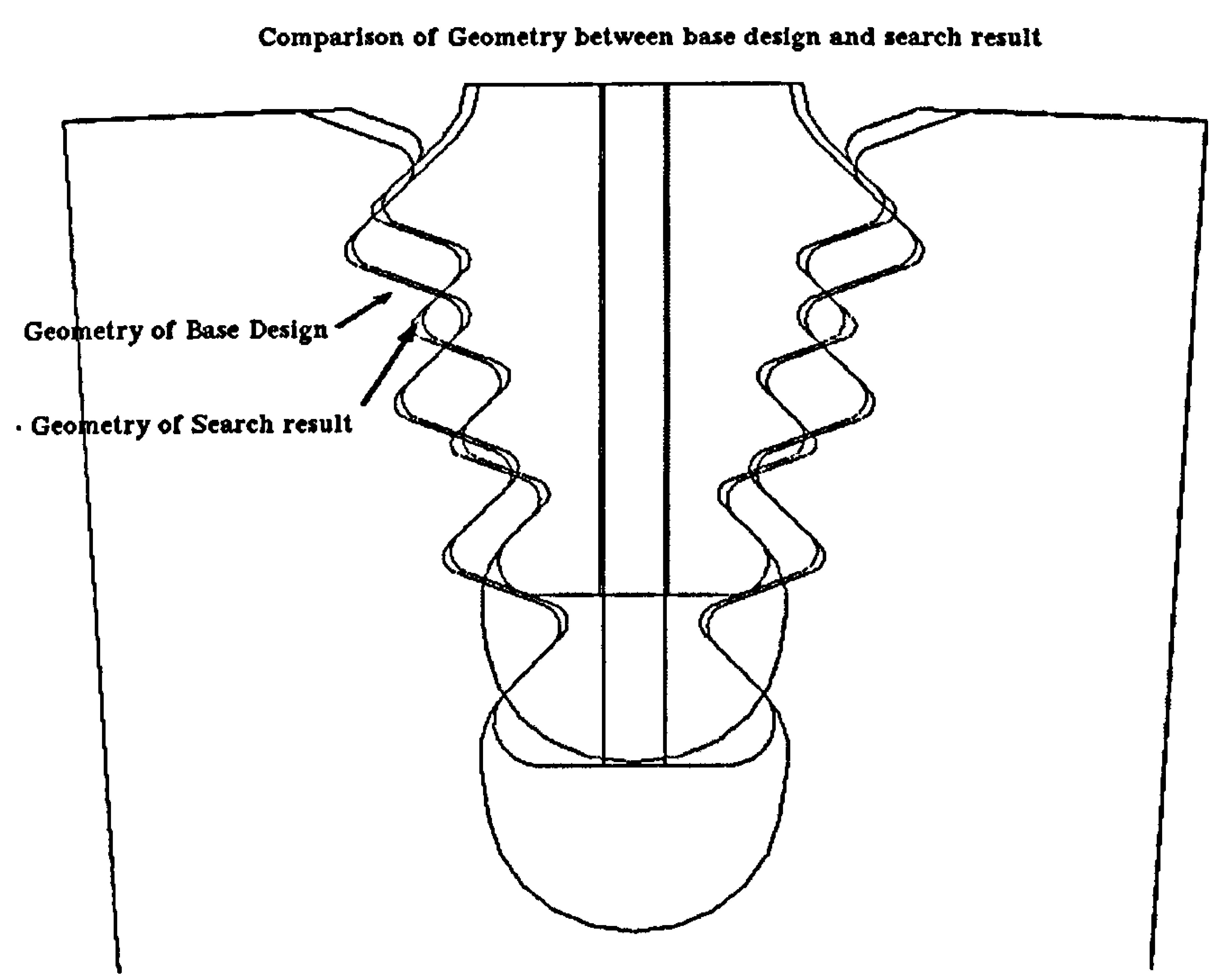


Figure 4.6: Comparison of Geometry between base design and search result for firtree frontal area reduction

Table 4.6 Optimisation results for minimising maximum disc notch stress

Variables	Start Values	Result Values
Root wedge angle	33.55963 (deg.)	21.1507 (deg.)
Tooth pitch	2.50146 (mm)	2.8197 (mm)
Blade crest radius	0.5492 (mm)	0.7167 (mm)
Blade trough radius	0.5468 (mm)	0.7009 (mm)
Disc crest radius	0.3698 (mm)	0.6847 (mm)
Disc trough radius	0.3016 (mm)	0.3388 (mm)
Maximum notch stress	817.978 (MPa)	601.92 (MPa)
Firtree Frontal Area	337.64(mm^2)	374.51(mm^2)

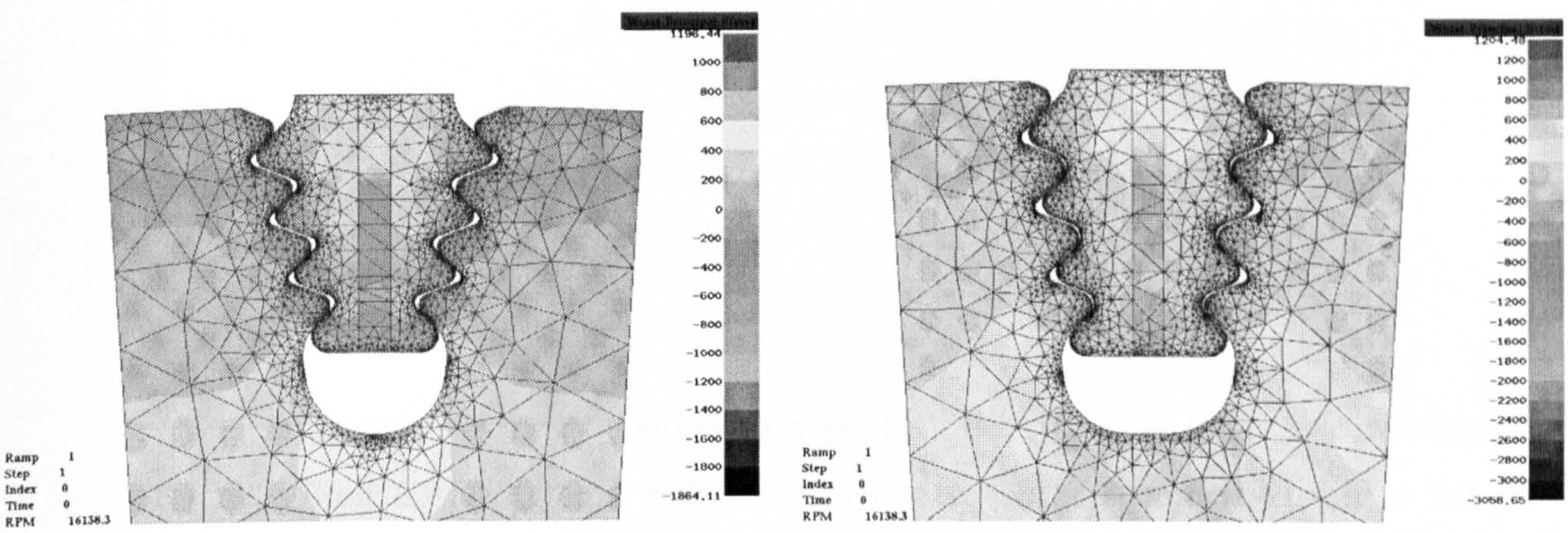


Figure 4.7: Tooth profile optimisation on search results from full-scale area minimization problem

(Left - results of initial optimisation as in Figure 4.6. Right: - Further result for minimizing stress)

Chapter 5

Exploration of New Shape Features Using NURBS

5.1. Introduction

The parametric geometry modelling capability provided by today's feature-based solid modelling CAD systems makes it possible to define geometry using a list of parameters, which then can be optimised against various geometric and stress constraints using efficient search strategies. However, there are a number of geometric shapes that cannot be modelled using existing geometric features, for example, the airfoil shape. Therefore free-form geometries could potentially be used to bring new features into shape design. In this chapter, Non-Uniform Rational B-Splines (NURBS) are introduced in the definition of tooth profiles in an effort to reduce the peak stresses in the notch region of the firtree root. The use of NURBS curves for notch fillet provides a greater flexibility in controlling the shape. Both the location of the control points and their weights can be varied to change the shape. A comprehensive treatment of NURBS can be found in Ref. 29.

In this work, non-dimensional coordinates of the control points are used to define the shape and are treated as design variables. NURBS curves are first described in the next section, followed by two definitions of notch fillet, one is defined using conic NURBS, the other is defined using cubic NURBS. Existing single arc and double arc fillets are then transformed into NURBS representations using these two definitions and used as starting points for optimisation.

5.2. Free-Form Geometry Modelling Using NURBS

Geometry modelling using polynomial and spline representations has been incorporated into most CAD packages, and these methods provide a universal mathematical approach to represent and exchange geometry in engineering applications.

The curves and surfaces represented in the ICAD system are, at their lowest level, represented by piecewise parametric polynomial functions. Many mathematical forms can be used to represent curves and surfaces. The most commonly used mathematical representation of curves and surfaces include Bezier and B-Spline functions, often referred to as B-forms. The Bezier representation of a curve has the following form

$$N(u) = \sum_{i=0}^{n} B_{i,p}(u) P_i \tag{5.2.1}$$

where P_i are control points and $B_{i,p}(u)$ are Bernstein polynomials of degree p. The $(n+1)$ control points form a control polygon and they are usually used as design variables. The definition of a Bezier curve requires only information on the control points. Although Bezier curves have many characteristics that are desired when designing a curve, they become oscillatory when the degree increases as the number of control points increases. In practice, several low-order Bezier curves are usually used to form a multi-segment curve, which is called a B-spline. A B-spline can be represented by

$$N(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{5.2.2}$$

where $N_{i,p}(u)$ is the i-th B-spline basis function of degree p. Unsurprisingly, B-splines have all the desirable properties of the Bezier representation. The definition of a B-spline requires not only the control points, but also a knot vector of length $m+1$ $(m = n + p + 1)$. The only drawback of the B-spline is its inability to represent the class of geometry represented by implicit conic sections. This results in the extension of B-spline to non-uniform rational B-spline (NURBS), which can be used to represent almost any shape.

A NURBS curve is defined from a set of control points, a weight vector and Bernstein basis functions as

$$N(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \tag{5.2.3}$$

where w_i is the weight correspondent to the control point P_i. For the design of a NURBS curve, these additional control parameters can be used as design variables along with the coordinates of the control points.

There are a number of properties that have resulted in NURBS being considered as a desirable mathematical representation in many CAD systems, and most CAD systems now support NURBS not only as their internal shape representation, but also provide an interface for NURBS modelling. In terms of shape optimisation, perhaps the two most useful characteristics are their strong convex hull property and their local control property. The strong convex hull property means that the curve is contained in the convex hull of its control points; in fact, the control polygon constitutes its approximation, therefore, the choice of the control points as design variables has clear geometric meaning. The local control property means that moving P_i only changes the curve in a limited interval; in contrast the change of the position of one control point affects the whole of a Bezier curve.

NURBS thus promise to be the future geometry standard for free-form curve and surface representations for geometry processing applications within CAD/CAM and scientific visualisation, solid modelling, numerically controlled machining and contouring. For example, a cubic NURBS curve defined using 4 control points is illustrated in Figure 5.1. The control of the curve, achieved by changing the control point positions and weights, are of interest for structural shape optimisation. The effect of modifying the weight of point P_2 is also shown in Figure 5.1.

The motivation behind the introduction of NURBS into the definition of the tooth profile comes from the idea that it may reduce the stress concentration around the notch regions of the firtree root while maintaining a constant contact length between the blade and disc. The first model considered is simply an extension of an existing traditional single-arc model, referred to as the conic fillet model. The single-arc fillet is first transformed into NURBS representation of degree two with the same shape. The control points instead of the radius are then used as design variables.

5.2.1. NURBS notch fillet of degree two – Conic fillet

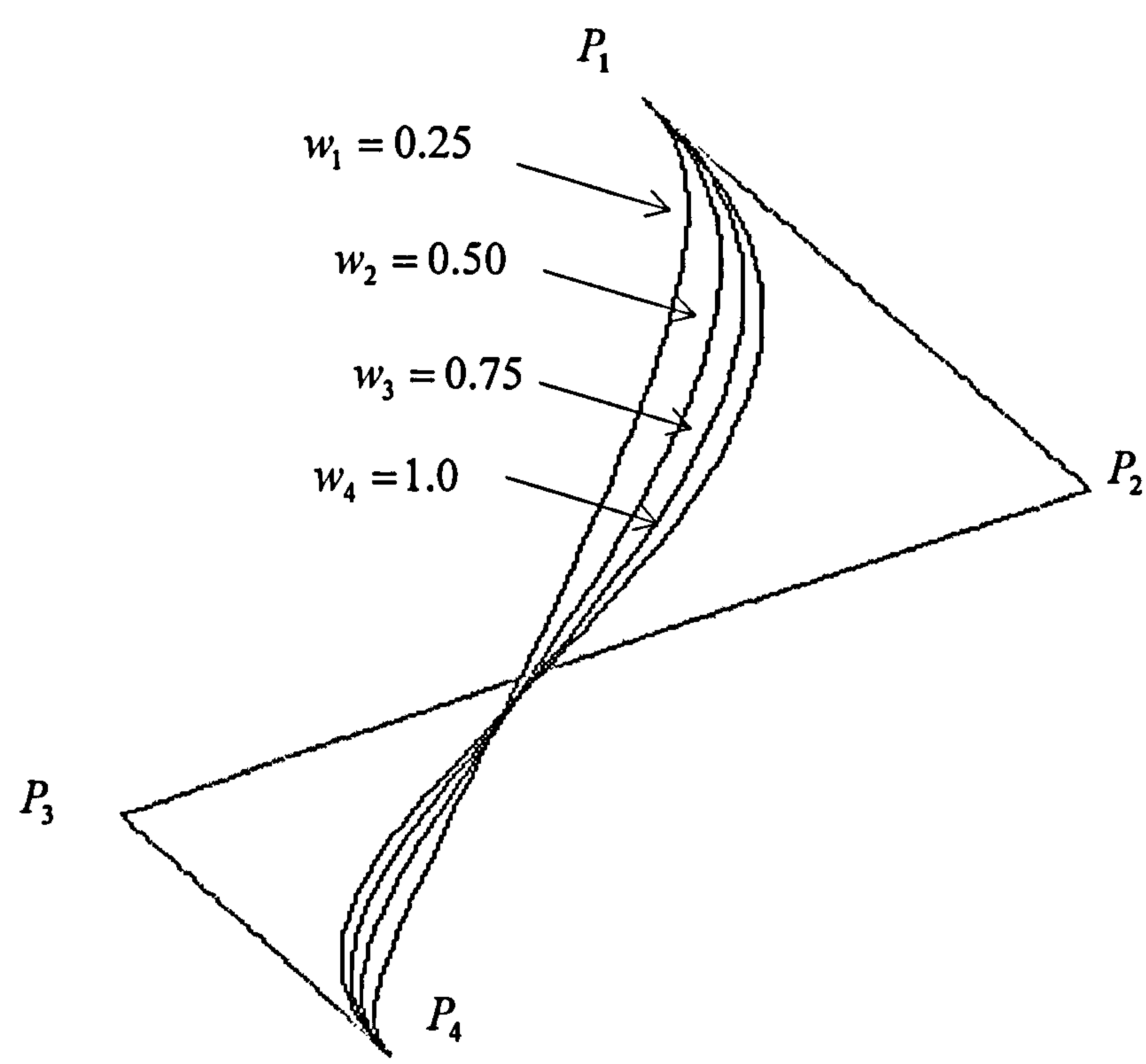


Figure 5.1: A NURBS curve and effect of changing the weight w of point P_2 on the shape

The simplest NURBS-arc of degree two can be defined using three control points, as shown in Figure 5.2 (defined by points P_0^0, P_1^0, P_2^0). The complete single tooth profile defined using this three-point NURBS is shown in Figure 5.3. Instead of using absolute coordinates, the non-dimensional coordinates are used here to define the position of the control points. This definition of the tooth profile is used as a starting point. Now moving

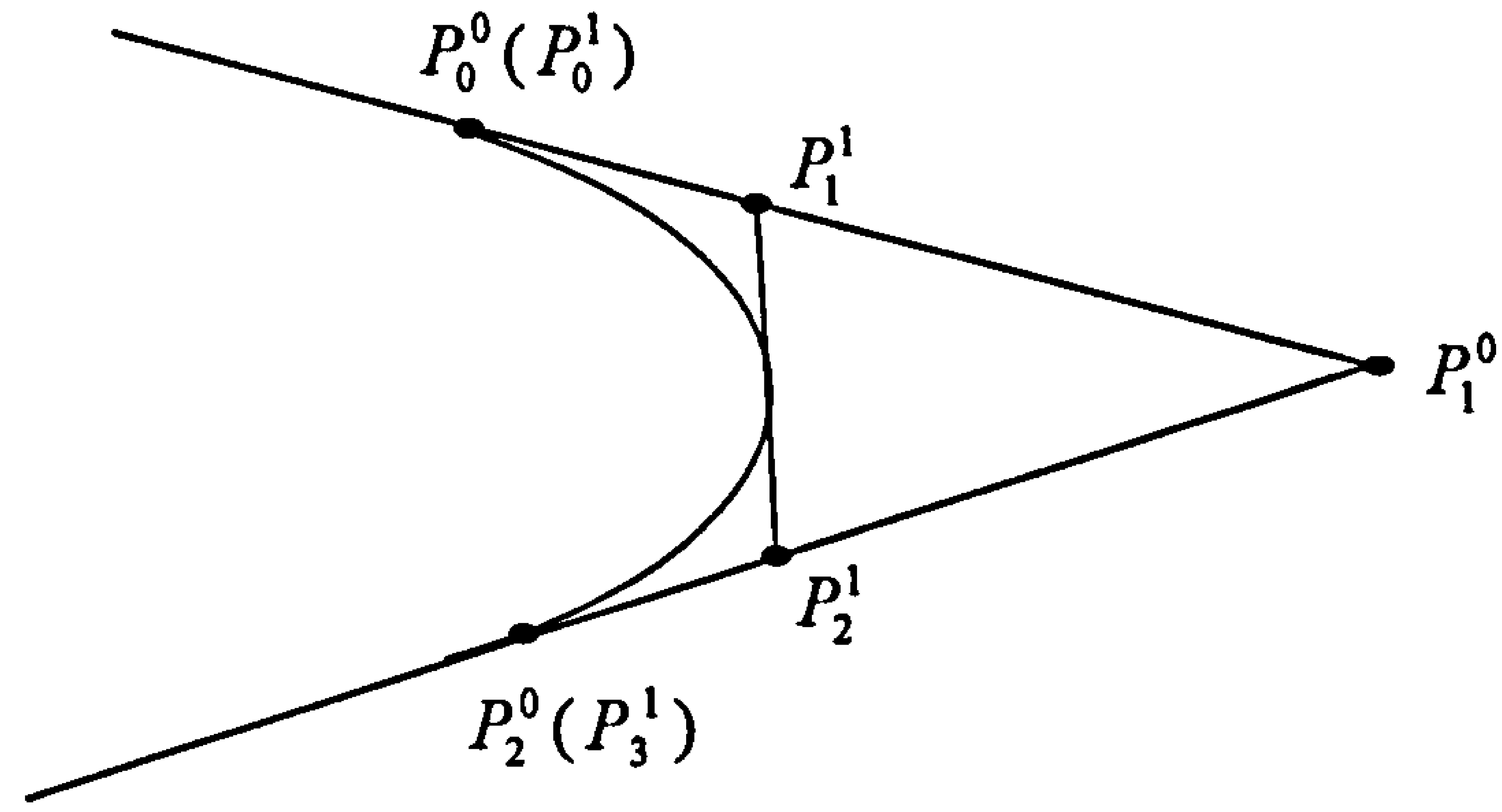


Figure 5.2: NURBS representation of single-arc fillet using ¾ control points

the middle control points will generate a series of conic curves, while modifying the weight associated with the middle control point will generate a different series of curves. However, the flexibility provided by this conic definition is still limited. Therefore, instead of using the simplest NURBS arc, a more general form of degree three involving four control points was adopted in this study; (defined by $P_0^1, P_1^1, P_2^1, P_3^1$), see Figure 5.2.

5.2.2. NURBS fillet of degree three – Cubic fillet

A double-arc fillet can be transformed into a NURBS representation using seven control points if the component arc is defined by a cubic NURBS, and the points are properly positioned, as illustrated in Figure 5.4, Describing the double-arc using NURBS instead of two radii brings much greater flexibility. The choice of a cubic curve is also consistent with the fact the most CAD systems use cubic curves internally to represent various geometries.

Describing the positions of the control points becomes difficult if the actual coordinates of the points are used, especially when each tooth profile is varied

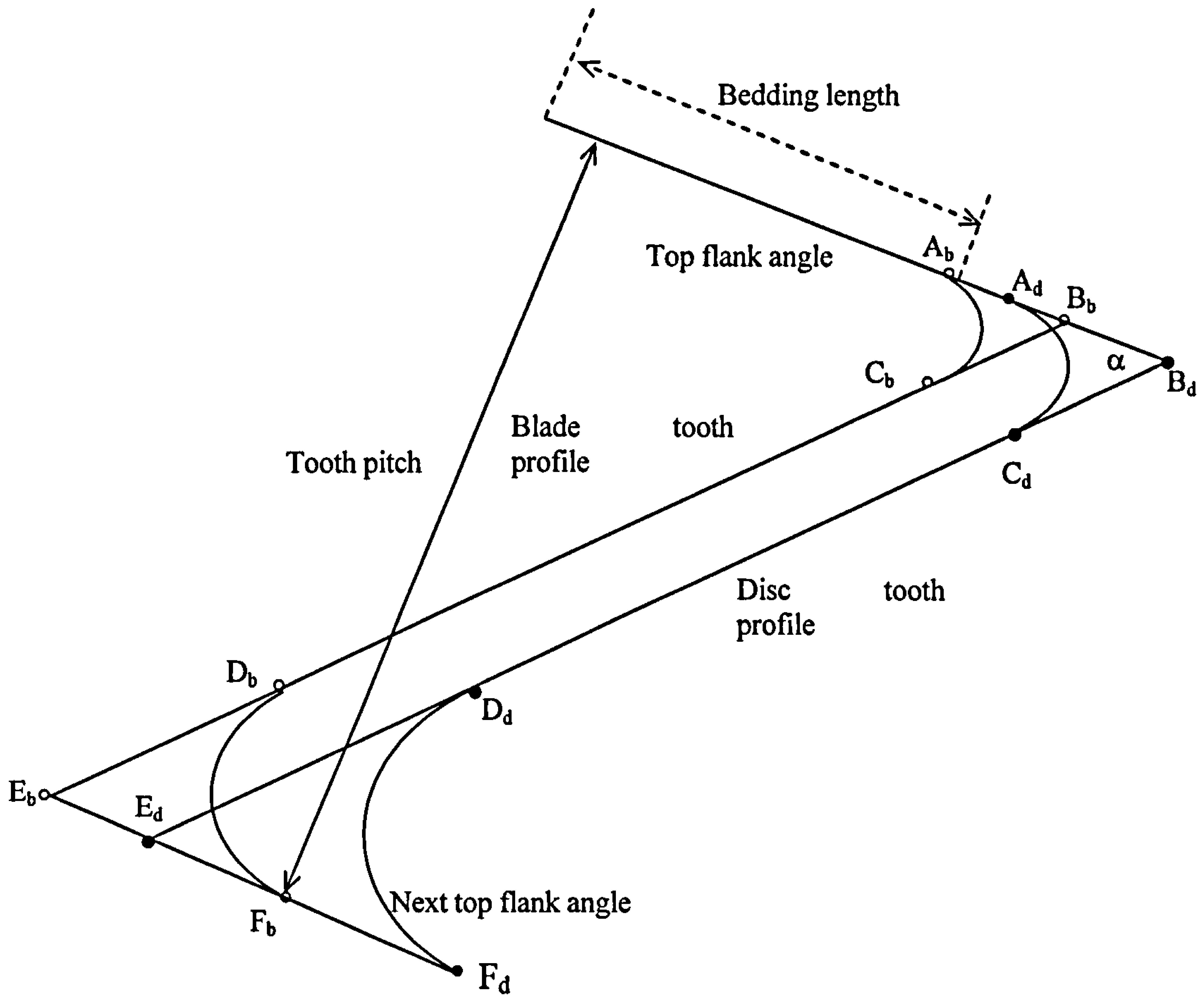


Figure 5.3: Notch conic fillet design using three control points

independently. Even for a uniform tooth profile, defining one tooth will involve 14 coordinates for these 7 points, and it is very difficult to manipulate these coordinates to ensure that an acceptable geometry is produced. Considering the convex property of NURBS curves and the fact that the tooth profile should always lie within the polygon prescribed by the points P_0^0, P_1^0, P_2^0 shown in Figure 5.2, a number of non-dimensional coordinates are used to define the position of the control points, as described next.

The point P_0 will be defined by the bedding-length (i.e., the end of the straight line segment) as illustrated in Figure 5.3, which is the main quantity affecting the crushing stress. The point P_6 is the end-point of the curve, essentially the point C_d in Figure 5.3 in the case of the disc profile. The point P_1 and P_5 are defined using two ratios ${P_0P_1}/{P_0O}$

and ${P_5P_6}/{P_6O}$, respectively, as shown in Figure 5.4. The point P_3 is defined by two non-dimensional quantities h,d as shown in Figure 5.4 (right) using points P_1 and P_5 as reference points. The points P_2 and P_4 are defined in a similar manner using P_1,P_3 and P_3,P_5 as reference points, respectively. This approach of defining control points provides better control as well as sufficient flexibility. Continuity requirements can then be built in to reduce the number of independent coordinates; in this case, the three points P_2,P_3,P_4 should form a straight line to maintain at least G^2 continuity of the curve at

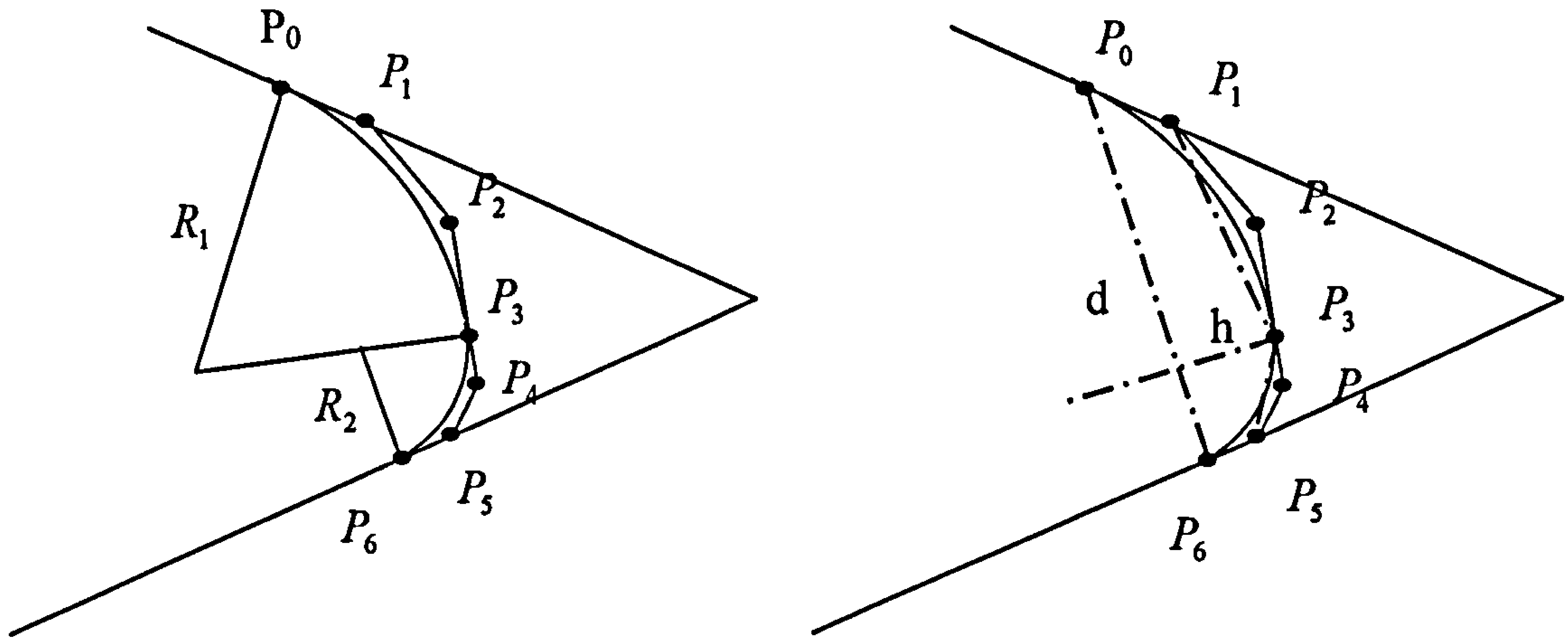


Figure 5.4: NURBS representation of double-arc fillet using 7 control points and its defining coordinates

point P_3 (two curves share the same curvature at the both sides of the junction). Another benefit of using non-dimensional coordinates is that the range of variations for the variables can be easily determined, therefore providing bound constraints for the design variables in optimisation.

To define a tooth profile using these 7 control points thus requires eight independent variables. The total number of parameters to be used in optimisation will be several times more if each tooth has a different profile of course. This definition method provides better control over the range of the variables without sacrificing the flexibilities offered by using free-form shape modelling.

5.3. Optimisation of new shape features using NURBS

The design of an optimum fillet shape not only depends on the methods employed to define the geometry, but also the loading conditions used, as a turbine blade is subjected to varying ratios of combined tensile and bending loadings. It also depends on the particular fillet under consideration. It has been shown by both experimental methods and finite element analyses that stress distributions for each fillet are different.[23,24]

The geometries that have been considered in the design of optimum fillets include circular arcs, elliptical fillets, combination of circular arcs and straight lines, and compounded circular fillets.[22,23,25] However, all these definitions lack the freedom that would be required to obtain an optimum shape in terms of reducing the peak notch stresses. Two different free-form definitions using NURBS are introduced here to investigate the potential benefits for shape optimisation. The first is conic NURBS of degree two; the second is cubic NURBS of degree three. Geometric definitions have been described in the previous section. Geometric parameters used in defining the conic and cubic fillet are listed in Tables 5.1 and 5.2, respectively.

Table 5.1 Geometry parameters and base values for tooth profiles with conic fillet

Variable	Name	Values	Units	Type
	Common parameters			
bdlin	Bedding length	1.06634	mm	Dimensional
	Blade Tooth Profile Parameters			
xdbeb	distance-between-db-eb	0.34615		Non-dimensional
xbbcb	distance-between-bb-cb	0.34615		Non-dimensional
	Disc Tooth Profile Parameters			
xabad	distance-between-ab-ad	0.08505	mm	Dimensional
xfdfb	distance-between-fd-fb	0.08505	mm	Dimensional
xfded	distance-between-fd-ed	1.03902	mm	Dimensional
xdded	distance-between-dd-ed	0.34615		Non-dimensional
xbdcd	distance-between-bd-cd	0.34615		Non-dimensional

There are two types of variables in the conic and cubic fillet geometry definition, one is dimensional, and the other is non-dimensional, as shown in Tables 5.1 and 5.2. The reasons for using non-dimensional variables have been explained earlier. The base values of the conic fillet correspond to those defining a single arc fillet, and the base values for the cubic fillet correspond to those defining a double arc fillet.

Each cubic profile is defined using seven control points. However, it is not desirable to use absolute coordinates to define the positions of these points as described in the previous section. Therefore, eight non-dimensional parameters are used to define a cubic fillet, as shown in Figure. 5.5.

Table 5.2 Geometry parameters and base values for tooth profiles with cubic fillet

Vars	Name	Value	Units	Type
	Common parameters			
bdlin	Bedding length	0.76272	mm	dimensional
zu0	distance-between-o-p1	0.20263		Non-dim
zv0	distance-between-o-p5	0.21940		Non-dim
zd1	distance-between-p1-p	0.54065		Non-dim
zm1	distance-tetween-p-p3	0.13859		Non-dim
ze1	parameters similar to zd1 and zm1, used	0.50971		Non-dim
zn1	to determine P_2 based on P_1 and P_3	0.09942		Non-dim
ze2	parameters similar to zd1 and zm1, used	0.67262		Non-dim
zn2	to determine P_4 based on P_5 and P_3.	0.16927		Non-dim
	Blade Tooth Profile Parameters			
zpbdb	distance-between-db-eb	0.346154		Non-dim
zpddd	distance-between-bb-cb	0.346154		Non-dim
	Disc Tooth Profile Parameters			
zabad	distance-between-ab-ad	0.085055	mm	Dimensional
zjdjb	distance-between-fd-fb	0.085055	mm	Dimensional
zqdjd	distance-between-fd-ed	1.039023	mm	Dimensional
zqbnb	distance-between-dd-ed	0.346154		Non-dim
zqdnd	distance-between-bd-cd	0.346154		Non-dim

Four of the eight non-dimensional parameters are defined as follows:

$$zu0 = {P_0P_1}/{P_0O} \quad (5.3.1a)$$

$$zv0 = {P_5P_6}/{P_6O} \quad (5.3.1b)$$

$$zd1 = {HP_1}/{P_1P_5} \quad (5.3.1c)$$

$$zm1 = {HP_3}/{P_1P_5} \quad (5.3.1d)$$

The remaining two pairs of parameters $(ze1, zn1)$ and $(ze2, zn2)$ are defined in a similar manner as the pair $(zd1, zm1)$. These parameters defining the local notch profile are then used as design variables in the local shape optimisation.

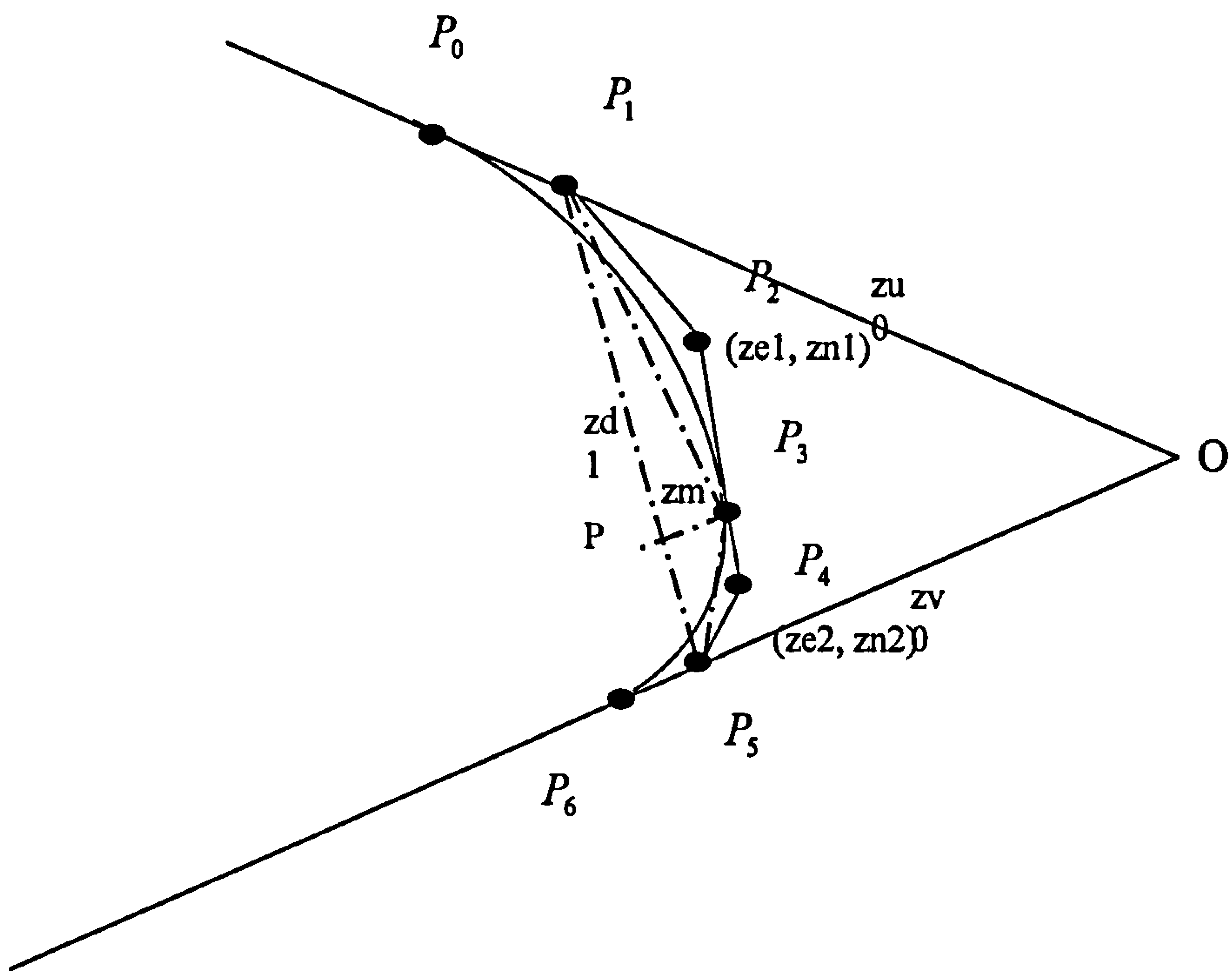


Figure 5.5: Illustration of non-dimensional coordinates for cubic fillet geometry

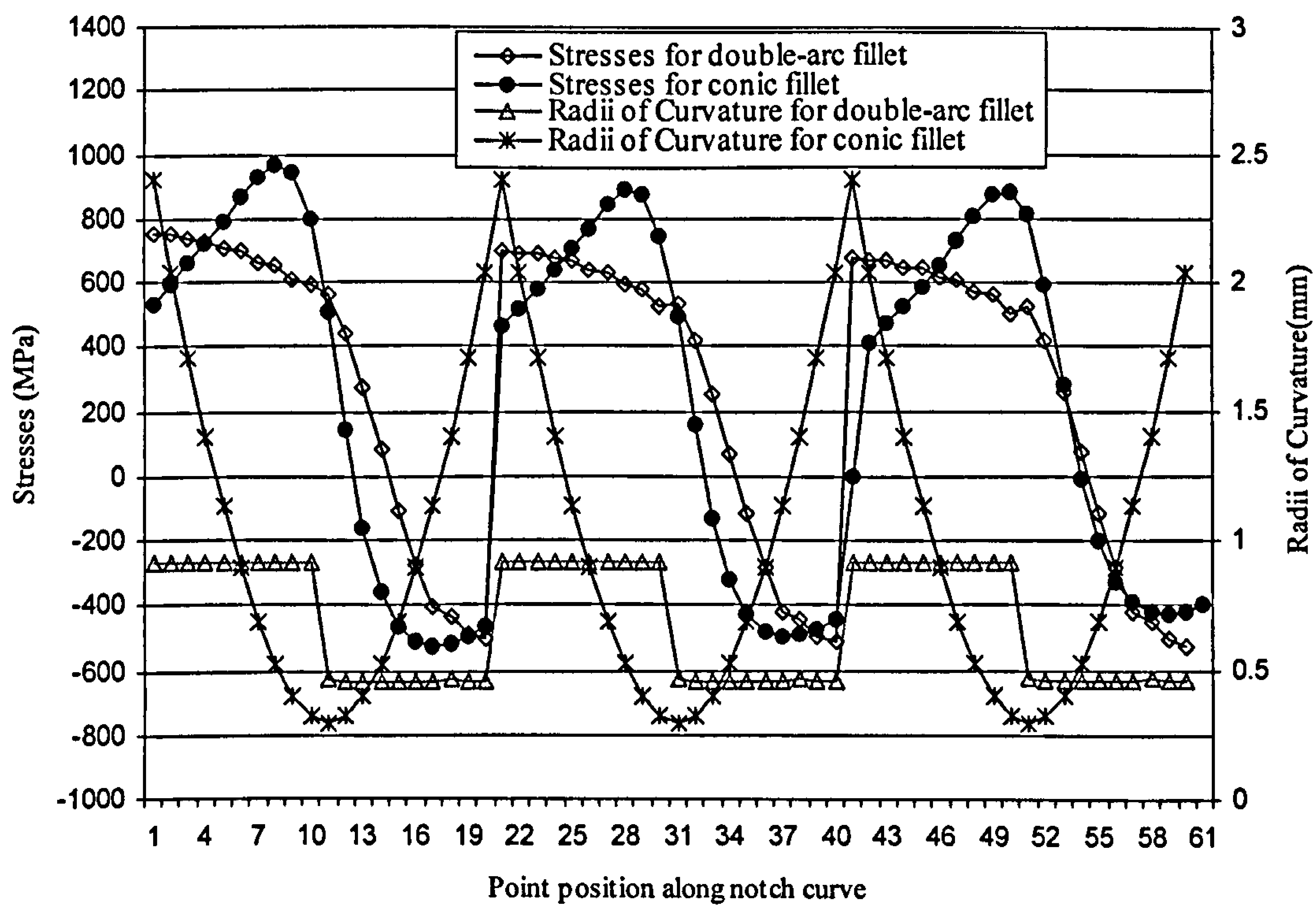


Figure 5.6: Distribution of notch stresses and radius of curvature along the notch curve.

Compared to the conic fillet, more design variables are required to define a cubic fillet. However, from initial analyses, the stress distributions along each tooth notch follows a similar pattern, this can be seen in Figure 5.6, where, the stress distributions along the notch region are shown. It is therefore a natural choice that the same non-dimensional values be used for all the teeth. Although this could be further relaxed to consider different tooth profiles, as a first step it is useful to focus on using the same set of non-dimensional parameters for all the cubic fillet profiles.

Replacing the single-arc fillet with a conic fillet brings explicit control over the bedding length and therefore the crushing stress. Then it becomes possible to decrease the crushing stress below the material criteria while at least maintaining the notch stress level at those seen in single-arc fillet designs, as shown in Figure 5.7. This figure is generated by plotting notch stresses against crushing stresses of all the solutions produced by a search using GA and stresses values are normalized against corresponding stress constraints. It can be seen, however, that it is difficult to reduce the notch stress as the conic fillet offers little more flexibility than the single arc fillet. Before moving on to the more flexible cubic fillet, a double arc fillet is next used to provide a basis for a general notch profile. The use of a double-arc fillet can further reduce the peak notch stress, and this can also be seen from Figure 5.7, where the objective function is set to

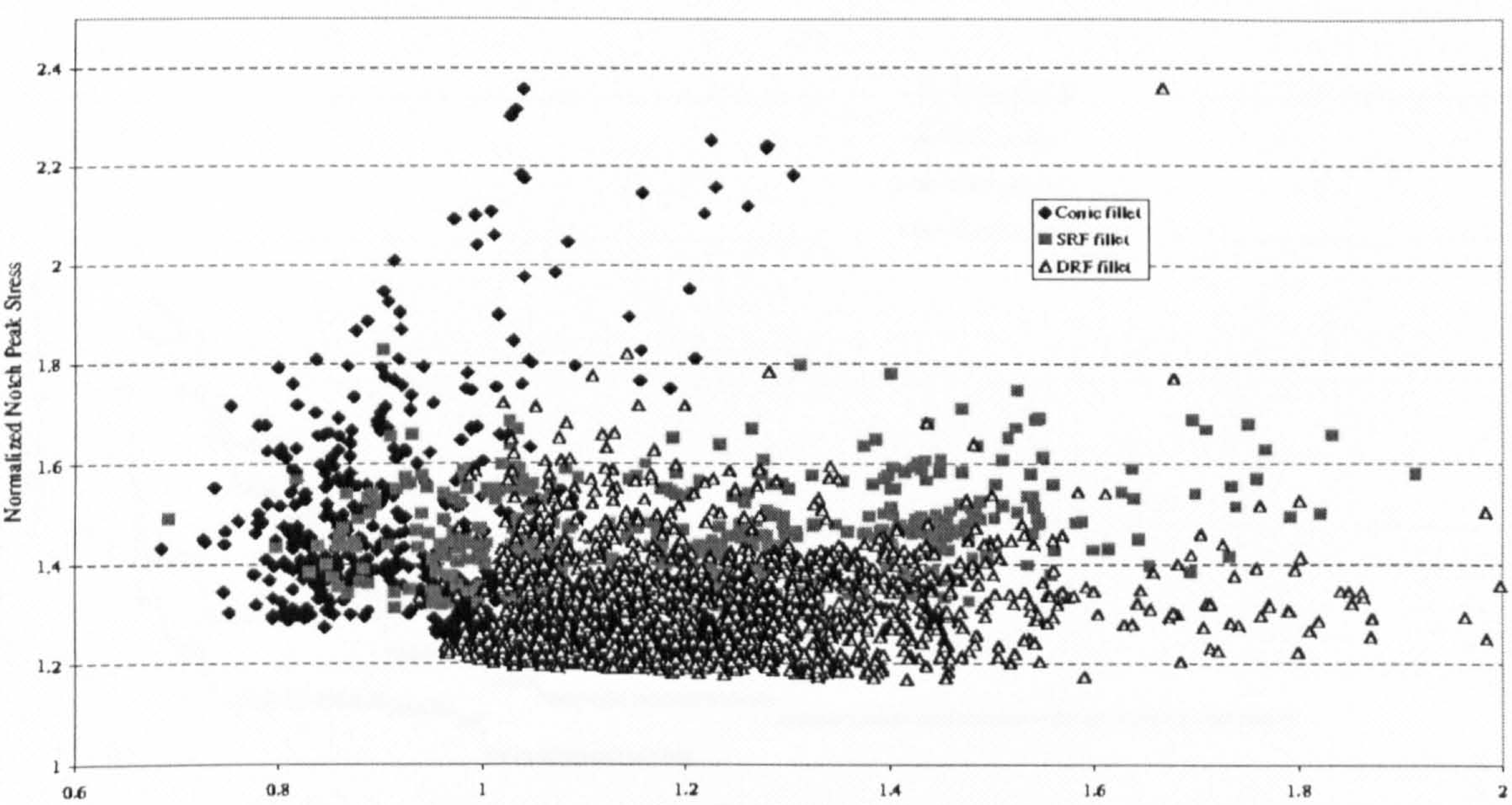


Figure 5.7: Normalized maximum notch stress against normalized maximum crushing stress. (normalization is defined as stress/material stress and all results considered during searches)

minimise the notch stress. It may be seen that both the crushing stress and notch stress are reduced compared to single-arc fillet, which is very difficult to achieve if the notch shape is restricted to a traditional single-arc design. The comparisons made in Figure 5.7 show that the use of NURBS can bring benefits and that a double-arc fillet provides a good basis for a NURBS model of the notch profile.

The geometry definition of a cubic tooth profile is provided in Section 5.2.2 using a list of dimensional and non-dimensional variables. The introduction of the cubic fillet provides further reductions to the disc notch peak stress, while all the other stresses can be maintained below the material allowable levels. This can be seen from the optimisation results shown in Tables 5.3 and 5.4. Definitions of design variables in these tables can be found in the geometry definition for different shapes in the previous sections. Convergence curves are shown in Figure 5.8 for the different types of notch fillet; the benefits of introducing NURBS fillets can be clearly seen. In table 5.3, comparisons between single-arc fillet and conic fillet show that although the conic fillet and the single-arc fillet start with the same shape, a lower notch peak stress can be achieved for conic fillet. Similar results can be seen in Table 5.4 when double-arc fillet and cubic fillet are used. The comparison is based on using different profiles for different

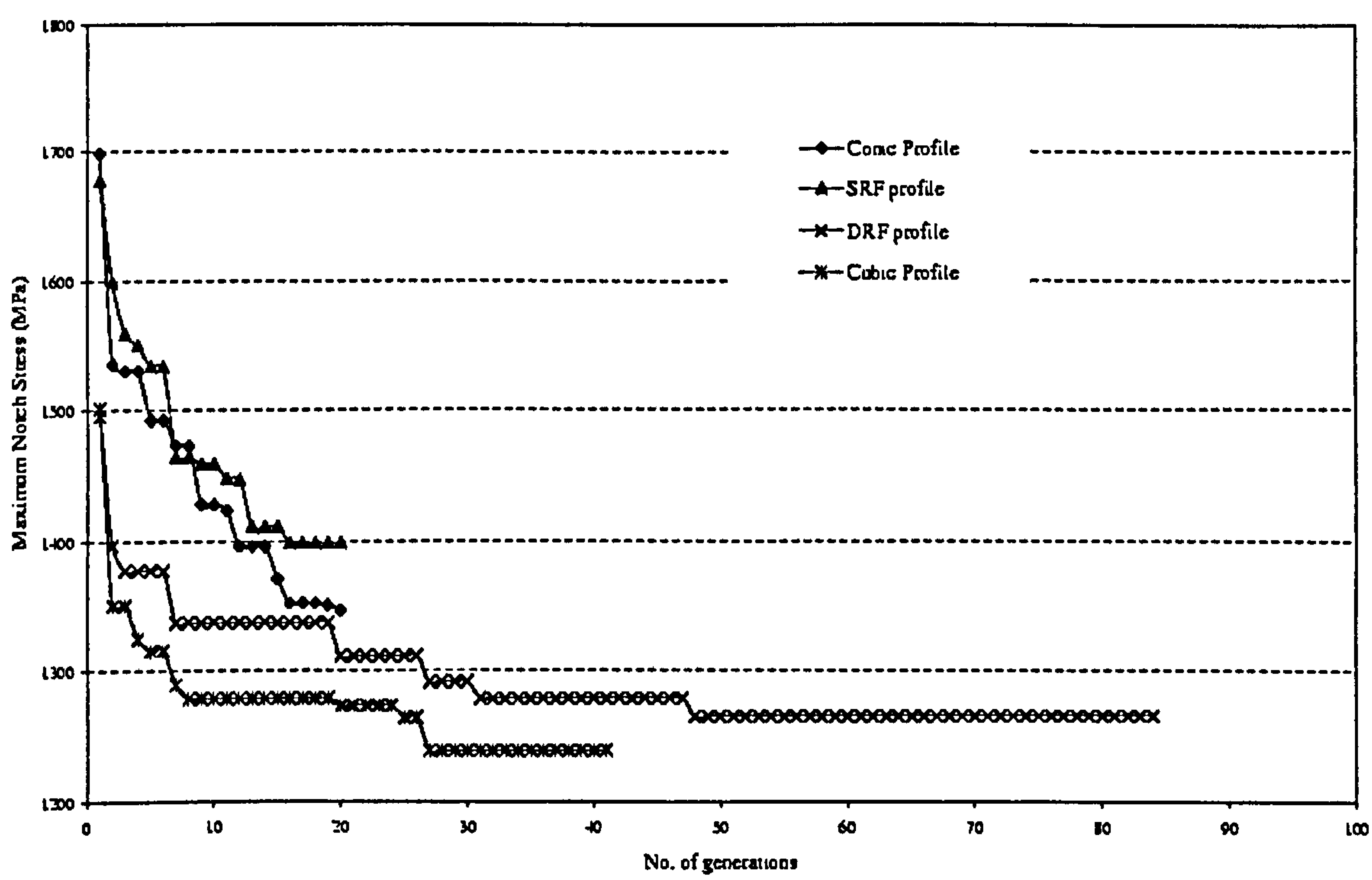


Figure 5.8: Convergence curves for different types of notch fillet using GA.

teeth for double-arc fillet and same profile for all the teeth for cubic fillet. A stress contour map for the notch region is shown in Figure 5.9, in which peak stresses before and after optimisation are noted for the cubic tooth profile design. However, whether or not the introduction of more flexible notch fillet will cause difficulties in manufacture or incur significant cost increases remains to be further investigated.

Table 5.3 Results of geometry parameters and constraints for single-arc and conic tooth

Tooth Type	Single-arc fillet Profile		conic Profile	
Design Variables	snfr	3.6711 (mm)	snfr	3.8938(mm)
	fsw	11.2(deg)	fsw	10.59(deg)
	btcr	0.9753(mm)	btcr	0.9964(mm)
	bglr1	3.8704(mm)	bglr1	3.1341(mm)
	bgur2	2.8216(mm)	bgur2	2.7167(mm)
	rwa	20.05(deg)	rwa	21.2899(deg)
	pitch	2.9634(mm)	pitch	2.9842(mm)
	rcrest	0.7388(mm)	bdlin	0.6486 (mm)
	fcrest	0.8799(mm)	xdbeb	0.2235
	rtrough	0.8384(mm)	xbbcb	0.4404
	ftrough	0.6814(mm)	xabad	0.1786(mm)
			xfdfb	0.0864(mm)
			xfded	0.8816(mm)
			xdded	0.4824
			xbdcd	0.4967
Objective (Maximum Disc Notch Stress)	1405.06MPa		1364.23MPa	
1.25<*R1/R2*	1.2226		1.1536	
0.30<*H/D*	0.4961		0.4475	
1.5<*PMIN*	2.9634		2.9842	
1.0<*BNP*	2.2711		1.9206	
1.0<*BNMIN*	2.1151		1.6157	
0.5<*AR*	1.1054		0.9834	
-1490<*NBR*<1490	1397.84MPa		1451.96MPa	
-1490<*NBL*<1490	1230.65MPa		1277.06MPa	
-385<*SB*<385	384.94MPa		384.82MPa	
-430<*SDR*<430	244.05MPa		238.20MPa	
-430<*SDL*<430	243.86MPa		288.77MPa	
-480<*CSR*<480	492.11MPa		479.39MPa	
-480<*CSL*<480	508.64MPa		441.64MPa	
-1400<*DBGL*<1400	624.20MPa		551.93MPa	
-1400<*DBGR*<1400	637.70MPa		567.39MPa	

Notes: Meanings of design variables for single-arc fillet can be found in Table 2.1; Meanings of design variables for conic fillet can be found in Table 5.1.

Table 5.4 Results of geometry parameters and constraints for double-arc and cubic tooth

Tooth Type	Double-arc fillet Profile		Cubic Profile	
Design Variables	snfr	3.8875 (mm)	bdlin	0.7298 (mm)
	fsw	10.677 (deg)	zqdjd	1.0163 (mm)
	btau	0.9924 (mm)	zjdjb	0.08492 (mm)
	btal	0.4038 (mm)	zabad	0.08505 (mm)
	bglr1	3.6399 (mm)	zpbdb	0.3035
	bgur2	2.276 (mm)	zpddd	0.3181
	rwa	22.54,24.636,20.267 (deg)	zqbnb	0.3005
	pitch	2.5336, 2.8579, 2.8203 (mm)	zqdnd	0.4132
	rctau	0.8565, 0.9876, 0.8526 (mm)	zu0	0.2044
	rctal	0.4601, 0.4107, 0.4606 (mm)	zv0	0.2543
	rtrau	0.2536, 0.4673, 0.5064 (mm)	zd1	0.7823
	rtral	0.9702, 0.9096, 0.9045 (mm)	zm1	0.1809
	fctau	0.5787, 0.5599, 0.4216 (mm)	ze1	0.6807
	fctal	0.8967, 0.8355, 0.9568 (mm)	zn1	0.06818
	ftrau	0.9894, 0.9850, 0.9653 (mm)	ze2	0.6392
	ftral	0.2233, 0.4054, 0.2964 (mm)	zn2	0.1815
Objective (Maximum Disc Notch Stress)	1265.4MPa		1239.2MPa	
1.25<*R1/R2*	1.5993		1.5965	
0.30<*H/D*	0.4658		0.4122	
1.5<*PMIN*	2.5336		2.6	
1.0<*BNP*	2.1749		2.208	
1.0<*BNMIN*	1.8578		1.6224	
0.5<*AR*	1.0123		0.8755	
-1490<*NBR*<1490	1516.5MPa		1544.3MPa	
-1490<*NBL*<1490	1309.6MPa		1340MPa	
-385<*SB*<385	384.98MPa		384.84MPa	
-430<*SDR*<430	305.57MPa		271.71MPa	
-430<*SDL*<430	301.23MPa		267.82MPa	
-480<*CSR*<480	481.64MPa		445.26MPa	
-480<*CSL*<480	478.08MPa		400.61MPa	
-1400<*DBGL*<1400	579.05MPa		575.76MPa	
-1400<*DBGR*<1400	597.3MPa		601.97MPa	

Notes: Meanings of design variables for double-arc fillet can be found in Table 2.2; Meanings of design variables for cubic fillet can be found in Table 5.2.

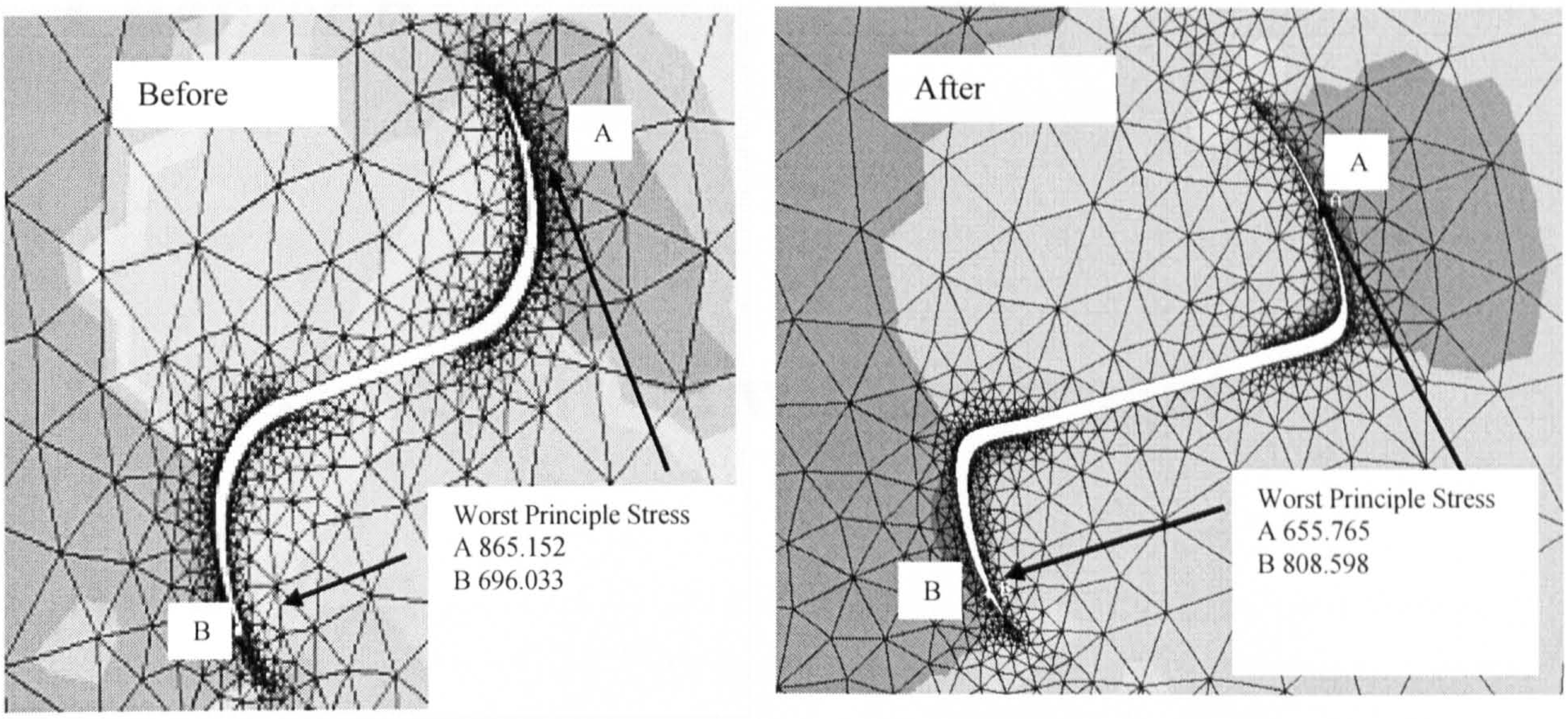


Figure 5.9: Contour maps for the notch region before and after optimisation
(The white "hole" at point A and B identify the position of peak stresses)

5.4. Conclusion

In this chapter, Non-Uniform Rational B-Splines (NURBS) is introduced in the definition of local notch profile in an attempt to reduce the peak stress without significant changes to the overall shape of the firtree. The introduction of NURBS increase the flexibility of the geometry definition, thus the possibility of free-form shape which further drops the peak stresses. Results obtained show that peak stresses can be reduced compared to conventional notch shapes consisting of circular arcs. The issues of manufacturability and additional cost that may incur need to be considered before such changes can be implemented in engineering practice.

Chapter 6

Genetic Algorithm with Surrogate Modelling

6.1. Introduction

The concept of using approximations is not new in the field of structural optimisation; it can be traced back to 1974 when Schmit and Farshi[66] published the concept of approximation techniques for structural synthesis. Since then, various approximation methods have been developed, such as the use of intermediate variables, (for example, the reciprocal of the original design variables),[9] force approximations for stress constraints,[67] and Rayleigh quotient approximations for frequency constraints,[68] to name a few. These techniques provide an efficient approach to obtain the responses at a reduced computational cost. For example, using member forces as intermediate variables, the optimum design of an eighteen-bar truss can be achieved using only eight detailed analyses.[69] However, these methods are difficult to incorporate into existing analysis capabilities and normally require problem-specific knowledge. It is even more challenging to adopt these methods for problems in the field of multidisciplinary optimisation. Therefore more general approximation schemes need to be developed to address the increasing cost associated with running the simulation codes for optimisation.

In recent years, there has been a growing interest in methods of using approximation models in optimisation, in which, the complex simulation codes are treated as black-box functions.[70-73] Data collected via evaluation of the expensive simulation codes at predefined design sites, often chosen by the method of design of experiment,[21] are used to construct mathematical surrogates. Various techniques for the construction of approximation models have been proposed. Perhaps the most popular techniques involve the polynomial approximation created by least square curve fitting to a set of data. Another line of methods is based on interpolation techniques, which are believed to be more suitable for data collected from running deterministic simulation codes. Methods of this type include radial basis function (RBF) developed in the field of neural networks,[74] stochastic-based methods, originated in the 1960s in the field of geostatistics, commonly referred to as Kriging or "DACE" (abbreviation of Design and Analysis of Computer Experiments) model, which is presented by Sacks, *et al.*[75] An efficient procedure for global optimisation using Kriging model was proposed by Jones *et al.*,[73] in which a branch-and-bound algorithm was applied to find point with potential maximum improvement for re-sampling. Other techniques using classifier systems and the concept of space mapping have also been proposed.[76,77]

An explorative comparison was made by Giunta *et al.*[78] between regression polynomial and interpolation models using test problems. A comparative study between neural network and response surface models was also provided by Daberkow *et al.*[79] using a preliminary aircraft design problem. Among various methods, RBF and Kriging were identified by Jin *et al.* to be able to produce better results than other methods under multiple modelling criteria.[80] Moreover, the Kriging method is statistically more meaningful and allows the possibility of computing error estimates for untried data points. One of the drawbacks of the Kriging method is the relatively high computational cost in estimating the hyper-parameters in the model (the undetermined parameters in the surrogate models), especially when a large number of responses are involved.

Apart from many different methods for constructing the surrogates, a number of different frameworks for the management of approximation models have also been proposed. It is argued that any effective framework for managing the surrogates in optimisation is always associated with particular optimisation algorithms. Therefore, in principle, there are as many possible frameworks as optimisation algorithms. In general, these frameworks can be categorized according to the algorithms in which they are used. There are two broad classes of optimisation algorithms, gradient-based method and non-gradient-based method in which, evolutionary algorithms are the most widely used. A rigorous framework was presented by Booker, *et al.*[81] for the use of surrogate models with direct search methods. Frameworks based on trust region and gradient-based search procedures have drawn most attentions in the past few years.[82-89] One of features possessed by this type of rigorous framework is that they all guarantee convergence to an optimum of the exact model. However, they work with non-linear programming techniques or direct search methods. The number of studies on how surrogate models can be used with evolutionary algorithms is relatively limited. However, the high computational cost associated with the successive application of evolutionary algorithms to complex, high-dimensional engineering problems is even worse than for gradient based searches, as typical evolutionary algorithms requires thousands of function evaluations to converge to a near optimal solution. Therefore, the use of surrogate models is believed to be crucial to the practical application of evolutionary optimisation methods to complex engineering systems.

Several attempts have been made in the last few years to tackle the problem of using cheap models with evolutionary search methods, particularly the Genetic Algorithms (GAs). Keane and Petruzzelli[90] employed variable-fidelity analysis models in the context of genetic algorithm for a transonic aircraft wing design problem. A procedure of using number of successive single-point approximation models with Genetic Algorithm (GA) was proposed by Nair,[91] where some physical knowledge was employed to construct the surrogate and a simple generation-delay method was used to control the use of exact models. Ratle[92] proposed a simple local convergence criterion to decide when the exact model should be resorted to in a procedure of integrating a Genetic Algorithm with Kriging models. However, this does not prevent the search from converging to a false or local optimum. A revised criterion was proposed by El-Beltagy[93] for a similar synthesis between GA and Kriging models, where a gradually reduced tolerance was used to control the switch between surrogate and exact models for each individual in the population. However, the specification of criteria values in the above methods depends largely on the users and may not be appropriate for the problem. Nevertheless, the requirement for some sort of re-sampling was identified in both papers as necessary to overcome the inadequacy of the surrogate models. A different type of effort was attempted by Liang,[94] where a hybrid search procedure was formulated with the evolutionary search working on the quadratic response surface constructed from many local optima obtained from local search results. This method essentially reduced the difficulties in building a global surrogate without changing the overall landscape of the exact functions. However, the quality of the final global surrogate model depends very strongly on the results of local search methods and use of evolutionary search methods in finding global optima in a simplified smooth landscape with few local optima may be questionable. Jin *et al.*[95] also proposed a framework for coupling Evolutionary Strategy (ES) and neural network-based surrogate models. Two types of evolution control methods were presented to decide the frequency at which the exact model should be used. The common weakness in the above methods is that neither the historical search data nor the convergence properties of the evolutionary search methods are fully utilized.

In this work, a Genetic Algorithm was coupled with Kriging surrogate modelling in order to reduce the computational cost without sacrificing the ability of GA finding global optimum for complex landscapes. Instead of using simple generation-delay criteria and user-specified tolerance control parameters, a 3σ principle derived based on the posterior variance estimate is used to suggest new sample points for exact re-evaluation. The new sample points are then inserted into an ordered database storing all historical exact solutions, and surrogate model is updated when these new points fall into the part that are used in the construction of the surrogate model. The following sections begin with a description of the Kriging method, followed by detailed discussion of the proposed framework for incorporating the Kriging model into a Genetic Algorithm. Two test functions are used to illustrate the effectiveness of the proposed framework.

6.2. Surrogate modelling

Let $Y(\mathbf{x})$ denote the true response of the system under study, and $\mathbf{x} = (x_1, \ \ldots, \ x_m)^T$ denote the vector of control variables. Sometimes the true response of the system can be represented in explicit mathematical form, but, in most cases, the knowledge of the system is incomplete or the model is too complex to represent using explicit mathematical functions, therefore a complex computer code is used to simulate the relationship between the responses and inputs. Whatever the case, observations can be made either through physical experiments or computer simulations at some values of the design variables. In this work, let us suppose that data has been collected at n points denoted by $\mathbf{x}^{(i)} = \left(x_1^{(i)},\ \ldots,\ x_m^{(i)}\right)(i = 1,\ldots,n)$, and the associated responses denoted by $y^{(i)} = Y\left(\mathbf{x}^{(i)}\right)$. Let $y(\mathbf{x})$ represent the approximation model. The relationship between the true response and approximation can be represented as follows:

$$Y(\mathbf{x}) = y(\mathbf{x}) + \Delta(\mathbf{x}) \tag{6.2.1}$$

The difference between the true response and approximated response, the total error or residual, is due to two types of errors, one is system error (bias error) denoted by $\varepsilon(\mathbf{x})$, which exists because of the incompleteness of the models employed. The second type is random error denoted by $\delta(\mathbf{x})$, which exists because of a number of reasons such as the effect of uncontrollable factors in the physical experiments, discretization errors typically encountered in the finite element analysis and computational fluid dynamics, and round off errors, etc. Therefore the total error is the sum of these two types of errors:

$$\Delta(\mathbf{x}) = \varepsilon(\mathbf{x}) + \delta(\mathbf{x}) \tag{6.2.2}$$

However, random errors can usually be controlled within certain level so that the output of a deterministic simulation code can be regarded as deterministic. Therefore the same set of inputs will produce the same outputs, this partly explains why a least square model does not always provide a reasonably good approximation to a deterministic computer simulation code. In this case, an interpolation model would be more suitable for creating approximations inasmuch as $Y\left(\mathbf{x}^{(i)}\right) = y\left(\mathbf{x}^{(i)}\right)$.

Among various techniques which interpolate the data, radial basis function (RBF) and Kriging were identified by Jin *et al.*[80] to be able to produce better results than other methods under multiple modelling criteria. The choice of Kriging techniques in this work is due to the fact that this method not only provides an estimate of the function values but also an estimate of posterior variance, which is used to control the frequency of re-sampling. A brief description of the Kriging model is provided below and detailed discussions can also be found in Refs. 73, 75 and 96.

The Kriging model is typically expressed as

$$y(\mathbf{x}) = \beta + Z(\mathbf{x}) \tag{6.2.3}$$

where β represents a constant term of the model, and $Z(\mathbf{x})$ is a Gaussian random process with zero mean and variance of σ^2. The covariance matrix of $Z(\mathbf{x})$ is given by

$$Cov\left(Z(\mathbf{x}^i), Z(\mathbf{x}^j)\right) = \sigma^2 R\left(\mathbf{x}^i, \mathbf{x}^j\right) \tag{6.2.4}$$

where σ^2 is the variance of the stochastic process and $R(.,.)$ is a correlation function between $\mathbf{x}^i$ and $\mathbf{x}^j$. Different types of correlation function can be employed as given in Refs. 73, 75 and 96. A commonly used type of correlation function can be expressed as

$$R\left(\mathbf{x}^i, \mathbf{x}^j\right) = \prod_{k=1}^{n} \exp(-\theta_k \left|x_k^i - x_k^j\right|^{p_k}) \tag{6.2.5}$$

where $\theta_k \geq 0$ and $0 < p_k \leq 2$ are the hyperparameters. Note that the above equation asserts that there is a complete correlation of a point with itself and this correlation deteriorate rapidly as the two points move away from each other in the parameter space. The choice of $p_k = 2$ would provide enough flexibility for modelling smooth and highly non-linear functions for most cases. The hyperparameters θ_k are estimated by maximizing the log-likelihood function given by

$$-\frac{1}{2}\left[n \ln \sigma^2 + \ln|\mathbf{R}| + \frac{1}{\sigma^2}(\mathbf{y} - \mathbf{1}\beta)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\beta)\right] \tag{6.2.6}$$

where σ^2 and β can be derived using the following equations once the θ_k are given

$$\hat{\beta} = \left(\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^T \mathbf{R}^{-1} \mathbf{y} \tag{6.2.7}$$

$$\hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{1}\beta)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\beta) \tag{6.2.8}$$

A numerical optimisation procedure is required to obtain the Maximum Likelihood Estimates (MLE) of hyperparameters. Once the hyperparameters are obtained from the training data, the function value at a new point can be predicted by

$$\hat{y}\left(\mathbf{x}^*\right) = \hat{\beta} + \mathbf{r}^T \mathbf{R}^{-1} \left(\mathbf{y} - \mathbf{1}\hat{\beta}\right) \tag{6.2.9}$$

along with the posterior variance $s^2\left(\mathbf{x}^*\right)$ given by

$$s^2\left(\mathbf{x}^*\right) = \sigma^2 \left[1 - \mathbf{r}\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2}{\left(\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}\right)}\right] \tag{6.2.10}$$

where $\mathbf{r}(\mathbf{x}) = \{R(\mathbf{x}, \mathbf{x}^1), \ldots, R\left(\mathbf{x}, \mathbf{x}^n\right)\}$ is the correlation vector between the new point $\mathbf{x}$ and training dataset. This quantity provides a good indication on the accuracy of the prediction at new points and will be used in our framework to decide whether further exact analyses are required.

To obtain an estimate of the accuracy of the predictions of the Kriging model, a leave-one-out cross-validation procedure can be employed as described in Ref. 73. The measure used in such a procedure is called the 'standardized cross validated residual' (SCVR) defined below

$$SCVR_j = \frac{y\left(\mathbf{x}^j\right) - \hat{y}_{-j}\left(\mathbf{x}^j\right)}{s_{-j}\left(\mathbf{x}^j\right)} \tag{6.2.11}$$

where $\hat{y}_{-j}(\mathbf{x}^j)$ and $s_{-j}(\mathbf{x}^j)$ denotes the mean and variance computed by (6.2.9) and (6.2.10) without using the *jth* training data. A good predictor would mean that the Gaussian process prior is appropriate for the dataset and majority of the $SCVR_j$ will scattered in the interval $[-3,3]$. Plotting the values of $SCVR_j$ against the predicted function values would also provide clues and solutions to problems that might exist in the model, for example, if there is any linear trend in such a plot, it is sometimes possible to improve the prediction by using log transformation.

Another very useful concept is the expected improvement discussed introduced by Jones *et al.*,[73] in which, the point with maximum expected improvements was found by using a branch-and-bound algorithm followed by re-sampling at that point and the re-construction of the surrogate. A simple 3σ principle is proposed and used instead of the maximum expected improvement in determining whether or not an evaluation using the exact model is necessary. This eliminates the need for an optimiser to find the point at which maximum expected improvement can be achieved. This principle is described below

$$\text{Evaluate } Y(\mathbf{x}^*) \text{ when } \hat{y}(\mathbf{x}^*) - 3s(\mathbf{x}^*) < \frac{1}{l}\sum_{j}^{l} y_j^{best} \quad (minimization) \qquad (6.2.12)$$

where $\hat{y}(\mathbf{x}^*)$ and $s(\mathbf{x}^*)$ are computed using (6.2.9) and (6.2.10), y_j^{best} represent the best l solutions in the dataset. Note that the right hand side reduces to $y_{\min}$ when $l = 1$ for minimization problems. How this principle can be used will be discussed in the following section.

It should also be mentioned that the basic Kriging model could further be extended to include derivative information as reported by Morris *et al.*[97] when derivatives are available either analytically or computed using automatic differentiation tools.[98] The availability of efficient adjoint methods for sensitivity computations makes this expansion more attractive than ever for complex simulation codes.

6.3. A Framework for managing the surrogates for Genetic Algorithms

One of the big challenges in using surrogate models in global optimisation is the balance between exploiting the surrogate and improving the accuracy of the surrogate. A commonly used strategy is to re-sample the point to which the surrogate has converged under certain criteria and to re-construct the surrogate using the augmented dataset.[93] A revised methodology would be re-sampling the point at which the expected improvement is maximized.[73] A common feature of these strategies for using surrogate with Genetic Algorithms is the use of an incrementally augmented dataset.

In this work, a general framework for managing the surrogates for Genetic Algorithms is proposed by the efficient and effective use of a simple 3σ principle and historical data. This framework is based on one of the basic features of Genetic Algorithms, that is, as the search proceeds, the population tends to be filled with more and more better solutions close to the global optima. There are two features that make this framework distinct from previously proposed ones.

- The use of a simple 3σ principle (6.2.12) eliminates the need for a search process for the point with maximum expected improvement, and;
- Ranking of the historical dataset allows more efficient and effective use of the results obtained from exact and usually computationally expensive codes.

The proposed procedure is illustrated in Figure 6.1. The algorithm starts with evaluation of the exact code on data points selected using design of experiment methods, for example, Latin Hypercube sampling.[21] The results are then archived in a central database according to a ranking of the fitness, and the top p or all of the solutions in the database are then used to construct the initial Kriging model. What follows is a typical process of the Genetic Algorithms with the fitness being evaluated first by surrogate model, if the 3σ principle applies, the exact expensive code is used and the result is then inserted into the database according to its fitness. The better solutions will come before the worse solutions and possibly into the top p of all solutions used in building the surrogate and will play a role in the update of the surrogate. The surrogate model is only updated when there has been a change in the top p solutions. It should be noted that the hyperparameters in the Kriging model are kept constant once found in the construction of the initial Kring model. As mentioned earlier, the number of similar solutions will increase as the GA proceeds, so the hyperparameters will have a less important role in the later stage of the GA search. This will further expedite the search as the estimation of hyperparameters itself involves the solution of an optimisation problem, which can be time-consuming. The update of the Kriging model will only involve the re-computation of the mean and variance using equations (6.2.7) and (6.2.8) based on the updated database.

The 3σ principle used is based on the fact that if the average fitness of top q designs lies outside the interval $\left[\hat{y}(\mathbf{x}^*)-3s(\mathbf{x}^*),\ \hat{y}(\mathbf{x}^*)+3s(\mathbf{x}^*)\right]$ computed at point $\mathbf{x}^*$, the probability of producing a better design at point $\mathbf{x}^*$ is very small. Two control parameters p (the number of design points used for surrogate modelling) and q (the number of exact results in the top of the ranked database used to compute the average fitness) are used to specify the number of design points in building the surrogate and in the 3σ principle. It is not difficult to understand the effect of these two parameters: increasing the parameter q will essentially lead to more new points falling into the 3σ interval and therefore more exact evaluations.

A real-coded Genetic Algorithm is used in this work, which is derived from the basic classes available from GAlib, an object-oriented class library developed by Matthew.[99] Instead of using a binary string to represent real numbers as in most commonly implementations of Genetic Algorithms,[16] the real coding offered a natural way of representing solutions in numerical optimisation problems. Each chromosome is an array of real numbers, e.g., the design variables. The use of real coding eliminates the need for coding and decoding process. Non-uniform crossover and random mutation operators are used in the current implementation of the GA and are briefly described below. Let $\mathbf{x}^i = \{x_1^i \ \ x_2^i \ \ ..., \ \ x_m^i\}$ and $\mathbf{x}^j = \{x_1^j \ \ x_2^j \ \ ..., \ \ x_m^j\}$ represent the parents in the population, the child $\mathbf{x}^c$ is generated by a BLX-crossover, which is defined as

$$x_k^c = \alpha\left(x_k^i - x_k^j\right), k = 1,...,m \tag{6.3.1}$$

Non-uniform mutation is defined as

$$x_k^c = \alpha\left(x_k^i - x_k^j\right), k = 1,\ldots,m \tag{6.3.2}$$

Finally, it should be pointed out that duplicate points are not archived to avoid problems in the computation of the inverse of the correlation matrix. And any type of implementation of population-based genetic algorithm could be used in this general framework for exploiting surrogate modelling.

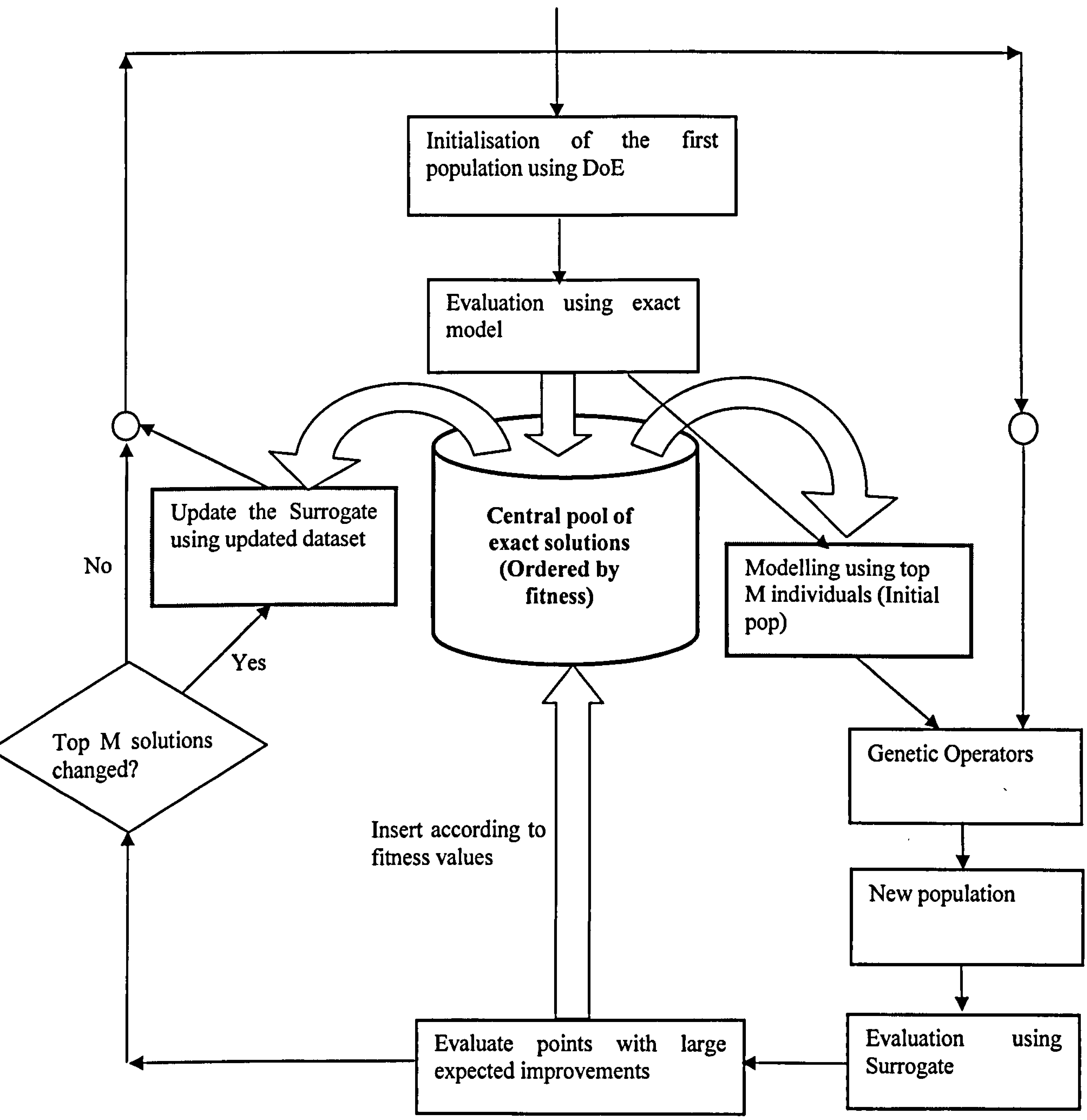


Figure 6.1: Procedure for incorporating Kriging model into Genetic Algorithms
(Broad arrows show data flows, narrow arrows show control flows)

6.4. Experiments on Benchmark Test functions

Two commonly used benchmark test functions are adopted here to test the effectiveness of the proposed framework. The first is an unconstrained 20D Rastringin function as defined by (6.4.1) and second is the constrained 20D Bump function introduced by Keane,[100] which is given by (6.4.2). Both of these two functions have large number of local minima and are usually difficult for standard optimisers to find global optima.

$$F_{Rastringin} = (10 * n) + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right) \tag{6.4.1}$$

$$F_{Bump} \quad \frac{abs\left(\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i) \right)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \text{ subject to } \prod_{i=1}^{n} x_i > 0.75 \text{ and } \sum_{i=1}^{n} x_i < 15n/2 \tag{6.4.2}$$

The bounds of variables for these two problems are $-5.12 < x_i < 5.12$ and $0 < x_i < 10$, respectively. The average convergence curves against the number of generations are shown in Figure 6.2 for the Rastringin function and in Figure 6.3 for Bump function. The results shown here were averaged over four runs for each test function. It can be seen that the number of exact analyses has been reduced to only one third of the original number of evaluations that would be required to obtain similar results using direct optimisation method. The standardized cross-validated residual is also shown in Figure 6.4 for the initial Kriging surrogate for the Rastringin problem. It is shown that a reasonably good approximation has been obtained. The proposed framework is applied to the local shape optimisation of the firtree problem in the next section.

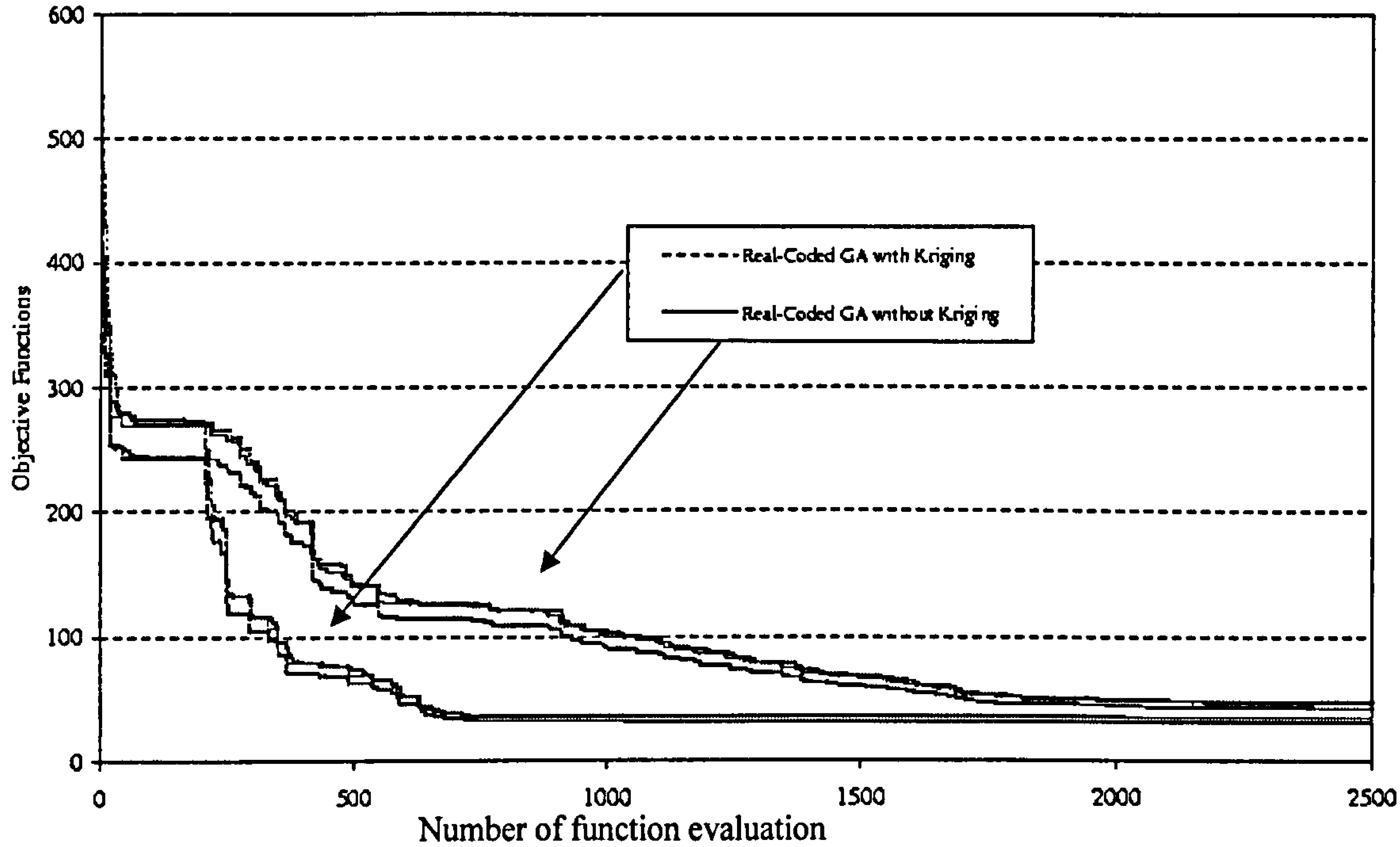


Figure 6.2: Genetic Algorithm with surrogate modelling on 20D Rastringing function

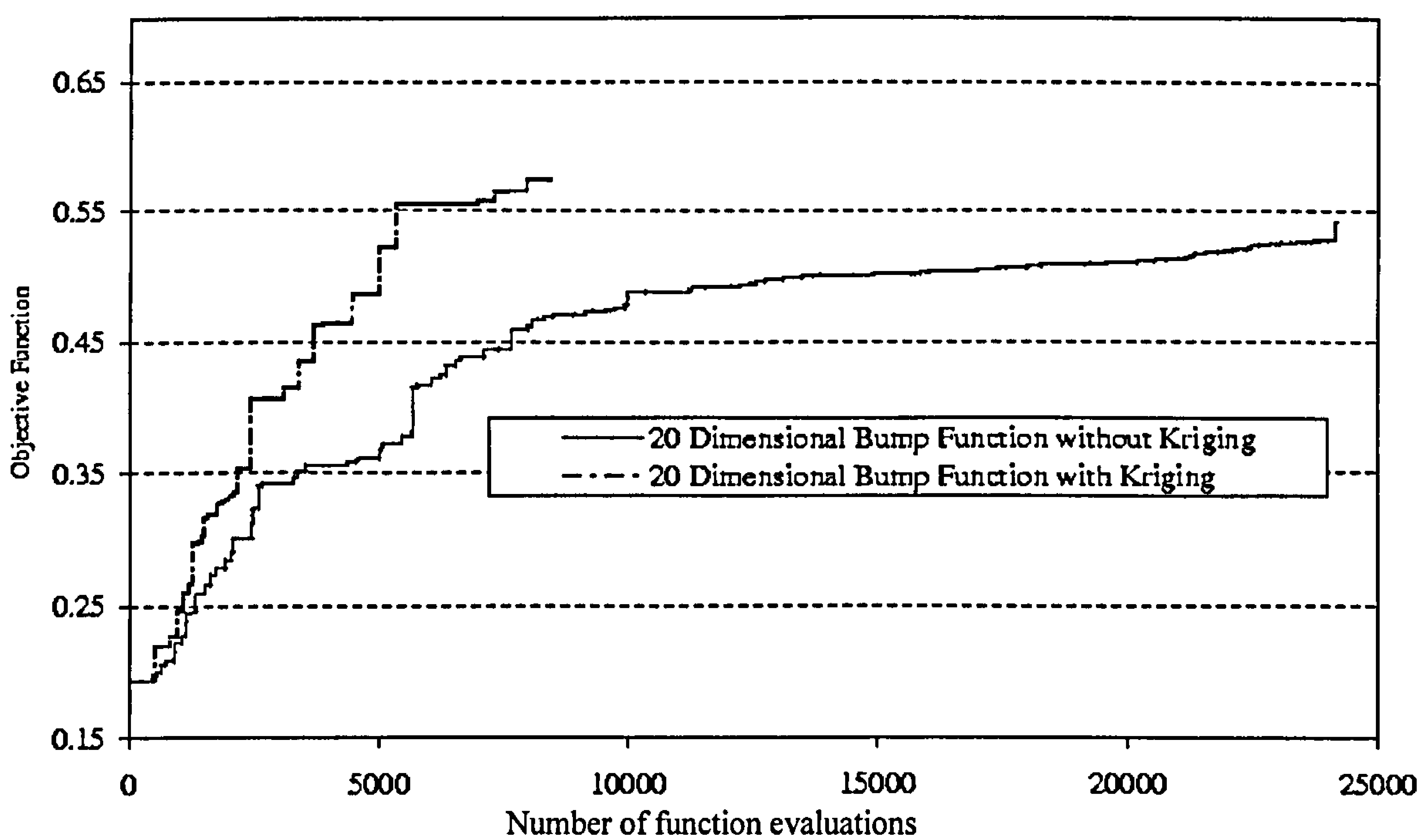


Figure 6.3: Genetic Algorithms with surrogate modelling on 20D Bump function

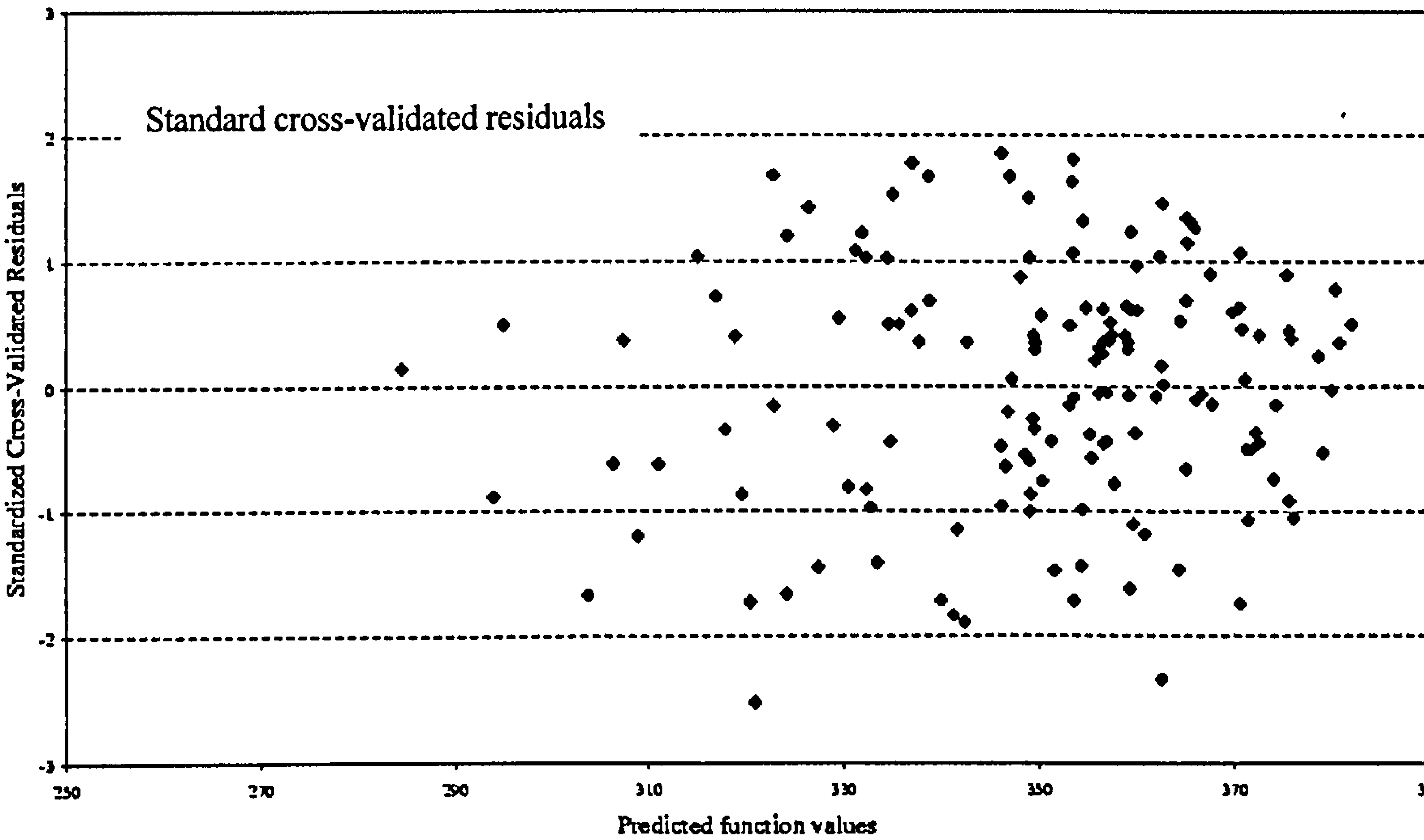


Figure 6.4: Standard Cross-Validated Residuals of initial Kriging model for Rastringing Test function

6.5. Local shape optimisation using Genetic Algorithm with surrogates

In this section, a local notch profile optimisation problem using cubic NURBS is chosen to illustrate the effectiveness of the proposed framework. The problem is to minimize the maximum notch stress when a cubic tooth profile is chosen. The definition of design variables can be found in Table 5.2. Results shown in Figure 6.5 are two convergence curves, one without the surrogate model, and the other with surrogate model. The result for surrogate-based genetic algorithm was averaged over three runs (also shown in Figure 6.5), while only one run was used for the result of the direct optimisation using genetic algorithm. A comparison of the final tooth profiles is given in Figure 6.6 and the values of all the design variables are listed in Table 6.1, along with objective function, and the constraints. The tooth shown in figure 6.6 is the top left tooth of the firtree where the peak disc notch stress occurred (a fixed skew factor of 1.75 is

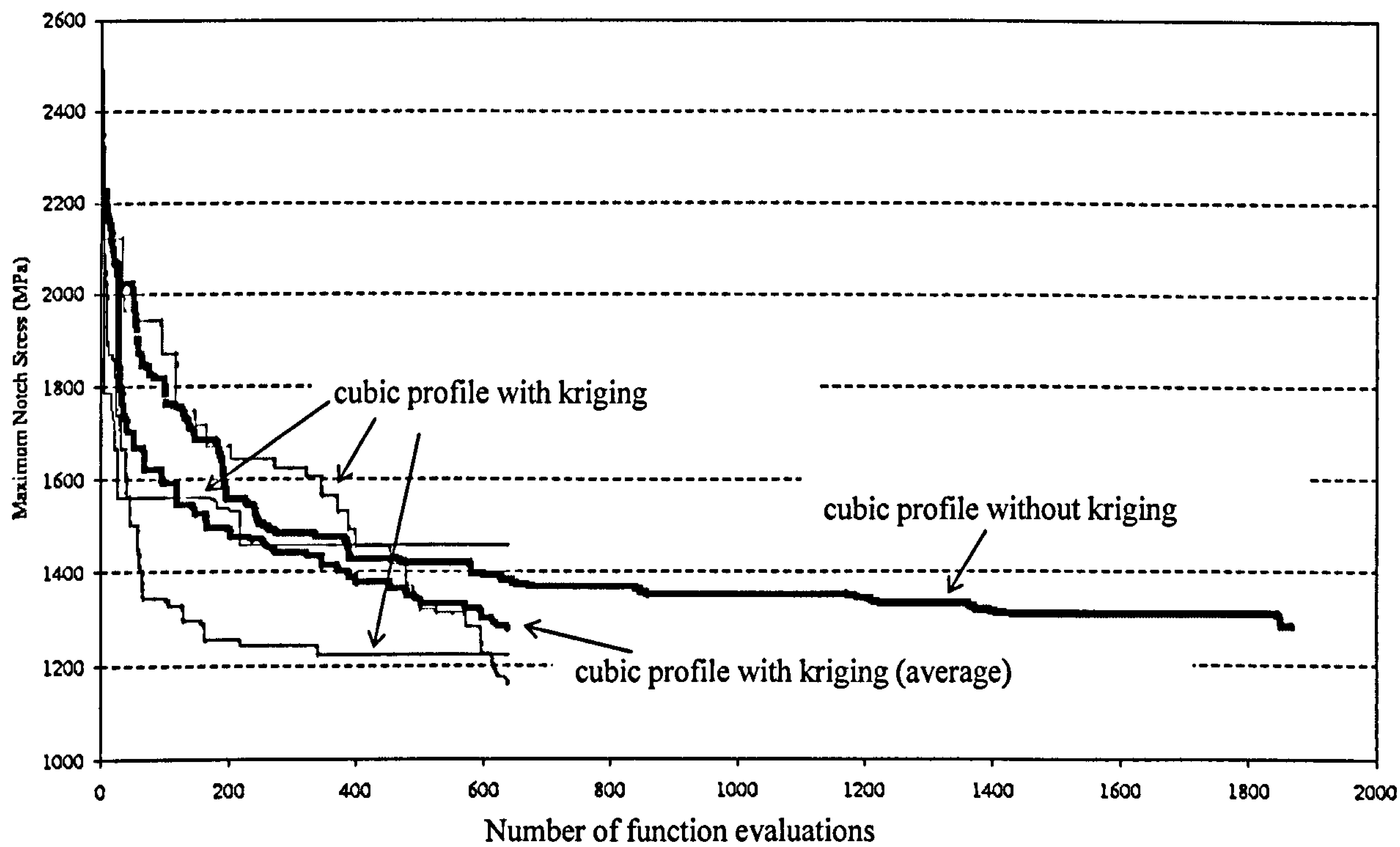


Figure 6.5:Genetic Algorithms with surrogate modelling on local shape optimisation

applied to the Worst Principal Stress shown in Figure 6.6). It can be seen that surrogate models can speed up the search process at some stage, on average similar disc trough profile can be obtained with only a portion of the number of exact function evaluations that would be required when direct search methods are used. Also, there is a difference in the blade trough profile and peak stresses. Blade notch peak stress using direct GA methods tends to higher than the one obtained with surrogates involved. However, these stresses are still within the material bounds imposed.

Table 6.1 Comparison between surrogate-based genetic algorithm and direct GA search

Optimisation of Cubic profile		Direct GA Search	Surrogate-based GA
Design Variables	bdlin	0.82004 (mm)	0.82004 (mm)
	zqdjd	1.0231 (mm)	1.0194 (mm)
	zjdjb	0.16243 (mm)	0.08272 (mm)
	zabad	0.085286 (mm)	0.051611 (mm)
	zpbdb	0.30193	0.30339
	zpddd	0.31805	0.32498
	zqbnb	0.3032	0.30056
	zqdnd	0.40156	0.30134
	zu0	0.29527	0.29368
	zv0	0.25416	0.25907
	zd1	0.78264	0.75724
	zm1	0.20705	0.20676
	ze1	0.78107	0.78576
	zn1	0.073659	0.068062
	ze2	0.73917	0.59421
	zn2	0.17996	0.18008
Objective (Maximum Disc Notch Stress)		1278.1MPa	1280MPa
$1.25<R1/R2$		1.5965	1.5965
$0.30<H/D$		0.41642	0.41642
$1.5<PMIN$		2.6	2.6
$1.0<BNP$		2.1809	2.1829
$1.0<BNMIN$		1.5888	1.5885
$0.5<AR$		0.8906	0.8906
$-1490<NB_{max}<1490$		1600.3MPa	1469.5MPa
$-385<SB_{max}<385$		384.84MPa	384.84MPa
$-430<SD_{max}<430$		274.71MPa	274.93MPa
$-480<CS_{max}<480$		541.69,465.79MPa	522.62,463.21MPa
$-1400<DBG_{max}<1400$		603.56MPa	603.88MPa

Notes: Meanings of design variables for cubic fillet can be found in Table 5.2.

6.6. Conclusion

In this chapter, a surrogate-based GA search scheme is proposed to tackle the problem of high computational cost when GA is applied to engineering optimisation problems involving expensive computations such as finite element analysis and computational fluid dynamics. A 3σ principle is adopted in the scheme to decide if the exact model needs to used, this eliminates the need for the optimisation effort in finding the point where expected improvement is maximized implemented in a previous procedure by Donald *et.al.*[73]. Results from both test functions and the problem of local shape optimisation of firtrees show that this scheme provide a good method to reduce the computational cost in GA-based optimisations.

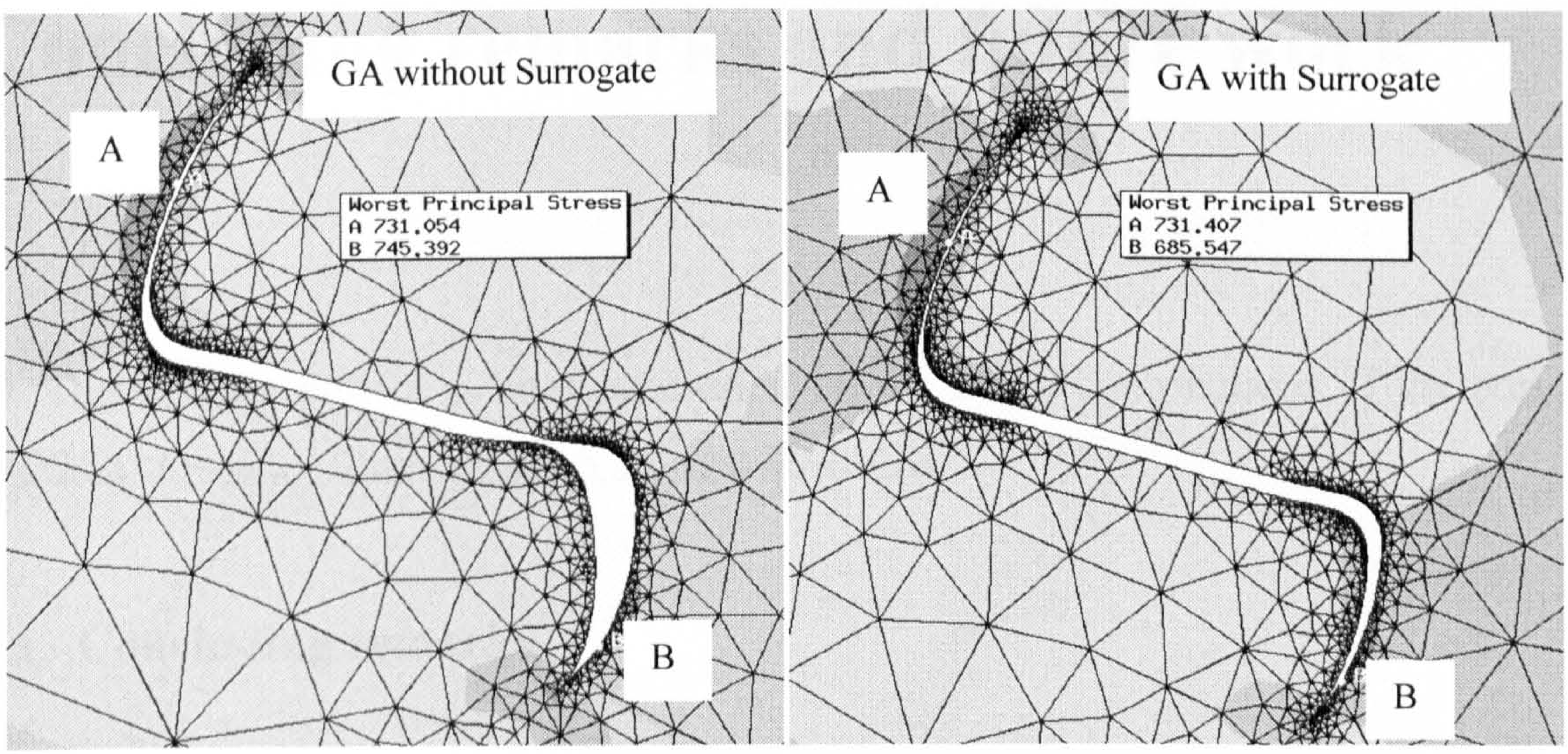


Figure 6.6: Comparison between results of GA search without and with surrogates

Chapter 7

Concluding remarks and future work

This chapter provide a brief conclusion on the major contributions of the present research. Areas for future research are also identified.

7.1. Concluding remarks

The use of CAD systems and various high-fidelity analysis tools in design optimisation requires the seamless integration of the various tools in order to carry out large number of analyses to reach an improved design. A fully automated system has been developed to couple the ICAD modelling tool and FE analysis, which not only provides a means to allow designers to review various design candidates in reduced time scales, but also can be incorporated into a search loop to find better designs using various optimisation schemes. This was successfully applied to the problem of turbine blade firtree design.

The rule-based generative modelling facility provided by the ICAD system allows effective changes to the topology as well as to the parameter values for a parametric CAD geometry. Incorporating such capabilities into the finite-element-based structural optimisation process has been shown to be an effective way to reduce design time scales and at the same time improve the quality of the final designs. The main contributions of the research work are three, the first concerns the system integration realised by the use of a tagged geometry and associated property data files. The automated system is then combined with a two-stage search strategy to find better shapes against a number of geometric and mechanical constraints. The second is the introduction of free-form NURBS in the fillet design for the notch region to reduce the stress concentration. This provides greater flexibilities compared to the classic straight-line and circular-arc design, and stress concentration can be further reduced. The third is the surrogate modelling scheme coupled with a Genetic Algorithm to reduce the computational cost by nearly two thirds of the original effort that would be required without resorting to surrogate models.

(1) A number of geometric rules derived from designers' experience and topology relationships between geometric entities are used to define the geometry and check the usefulness of a shape. Each entity is tagged with a unique name to allow the specification of boundary conditions, load, and domain information in the finite element code. This rule-based geometry model is capable of:

- dealing with different numbers of teeth and different sizes of teeth;
- evaluating geometrical constraints;
- automatically generating geometry dependent information as the topology changes

Two different types of search methods have been hybridised in such a way that a genetic algorithm is first used to locate promising areas in the whole design space defined by the boundary of the variables before a gradient-based procedures takes over on the good designs identified. This is based on the observation that Genetic Algorithms can provide fairly even coverage over the search space and results from GA search constitute promising starting points for more accurate local searches.

(2) A combined geometric parameterisation method using both basic geometric features and free-form shapes is presented and applied to the local shape optimisation of a turbine blade firtree. This is used to further improve the firtree design compared with traditional tooth profiles consisting of straight lines and circular arcs only. Non-uniform rational B-splines (NURBS) are used to model the tooth profile. The geometry is constructed from a number of geometric features and localized free form shape control points. The control points and related weights of the NURBS curves along with several

geometric feature dimensions are then chosen as design variables. Results showed that notch peak stresses can be reduced compared to the traditional straight-line/circular arc tooth design.

(3) The computational cost associated with design optimisation is usually very high when high-fidelity codes are used to evaluate the objectives and/or constraints. The situation deteriorates when evolutionary search methods such as Genetic Algorithms are used, as such methods typically require more function evaluations to converge. This problem is normally tackled with the use of surrogate models, typically a polynomial response surface. There have been a number of publications concerning the use of surrogate models with gradient-based methods and direct search methods. However, research concerning the use of surrogate models with Genetic Algorithms is less common. In this work, a Gaussian process based surrogate model, also known as Kriging is used because of its ability to provide estimation on the prediction accuracy. The usual process of repeatedly updating the hyperparameters when new data becomes available, which has been used in previous work, is removed by making effective use of the convergence property of the Genetic Algorithms. Results on test functions and also local shape optimisation problems of the firtree root have shown that this scheme can reduce the number of function evaluations while obtaining the same level of design improvements.

7.2. Future work

A number of potential future research directions has been identified based on results and experience obtained from the current work. They are listed below.

- **Robust manufacture-based geometry modelling**. Rule-based geometry modelling provides a way to effectively capture various items of knowledge. However, the set of rules is not unique in deciding the shapes, and it is believed that the introduction of manufacturing rules would improve the robustness of the model.
- **3D shape optimisation of the firtree root**. In this work, the 3D effect of skew angle on the peak notch stress was considered by using a scalar factor derived from comparative studies between 2D and 3D finite element models. These studies are

normally carried out for specific types of geometries. The validity of these factor may be in doubt for other types of geometries such as double-arc or NURBS. As the notch peak stress is a key factor affecting the life estimates of the turbine components, it would be beneficial to carry out 3D shape optimisation for the local shape.

- **Sensitivity studies and application of probabilistic methods in shape optimisation.** Decisions about the shape parameters should not only be judged by their impact on the weight and stress. Sensitivity studies should also be considered as some of the responses, such as peak stresses, have greater sensitivities to some of the shape parameters. Furthermore, manufacturing tolerances need also be considered in the shape design. Probabilistic methods would have a role to play in the shape optimisation, especially for the local shape changes.
- **Local Kriging models.** Kriging is based on correlations between existing points and any untried new point: the greater the distance between existing points and a new point, the smaller the effect of these points on the prediction at the new point. Therefore, a local Kriging model could be constructed using a small number of nearest neighbours instead of a global Kriging model. It is expected that the computational cost associated with building a local Kriging model would be much smaller as an optimisation procedure is typically required to estimate the hyperparameters in the model.
- **More complicated scheme for constructing and using surrogate models.** Various surrogate techniques have been proposed along with different strategies of using them. Radial basis function (RBF) and trust region schemes have shown much success when coupled with gradient-based techniques. However, the classic polynomial response surface method still has a role to play because of its simplicity and ease of use. Constructing a global polynomial response surface based on local extremes found by gradient-based methods could be used on problems when the computational cost is extremely high.

References

[1] Biedron R. T., Mehrotra T., Nelson M.L., Preston F.S., Rehder J.J., etc., "Compute as fast as the engineers can think!", NASA/TM-1999-209715.

[2] ANSYS Inc., http://www.ansys.com, 2002.

[3] Samareh, J. A, "Survey of shape parameterisation techniques for high-fidelity multidisciplinary shape optimisation", AIAA Journal, Vol. 39, No.5, pp.887-884. May 2001.

[4] Lanousin H. J. and Waggenspack W. N. Jr., "NURBS-Based Free-Form Deformations", IEEE Computer Graphics and Application, Vol. 14, No. 6, November 1994. pp 59-65.

[5] Robinson G. M. and Keane A. J., "Concise Orthogonal Representation of Supercritical Airfoils", Journal of Aircraft, Vol. 38, No. 3, pp 580-583, 2001.

[6] Lépine, J., Guibault, F., Trépanier, J.-Y. and Pépin, F., "Optimized Nonuniform Rational B-Spline Geometrical Representation for Aerodynamic Design of Wings", AIAA Journal, Vol. 39, No. 11, November 2001.

[7] Knowledge Technologies International, http://www.ktiworld.com.

[8] Song, W., Keane, A.J., Rees, J., Bhaskar A. and Bagnall, S., "Turbine blade Fir-tree root design optimisation with intelligent CAD and finite element analysis", Computers and Structures, 80, 1853-1867, 2002.

[9] Vanderplaats G.N., "Structural Design Optimisation Status and Direction", Journal of Aircraft, Vol. 36, No.1, January-February 1999.

[10] Papalambros, P.Y. and Wilde D.J., *Principles of optimal design: modeling and computation*, second edition, 2000, Cambridge University Press.

[11] Keane A. J., Design Exploration System: OPTIONS, 2002, http://www.soton.ac.uk/~ajk/options.ps.

[12] iSight, Engineous Software, 2002, http://www.engineous-software.com/index.html.

[13] Pardalos, P.M., Romejin, H.E. and Tuy, H., "Recent Developments and Trends in Global Optimisation", Journal of Computational and Applied Mathematics 124, pp. 209-228, 2000.

[14] Neumaier A., Global Optimisation, http://www.mat.univie.ac.at/~neum/glopt.html.

[15] Lewis R.M., Torczon V., and Trosset M.W., "Direct Search Methods: Then and Now", NASA/CR-2000-210125, ICASE Report No. 2000-26, 2000.

[16] Goldberg D.E., *Genetic Algorithms in Search, Optimisation and Machine Leaning*, Addison-Wesley. 1989.

[17] Kirkpatrick, S., Gelatt C. D. Jr., and Vecchi M.P., "Optimisation by simulated annealing", Science 220 (4598) pp. 671-680, 1983.

[18] Bäck T., Schwefel, H.-P., "Evolutionary computation: an overview", ed. T. Fukuda, T. Furuhashi and D. B. Fogel, Proceedings of 1996 IEEE International Conference Evolutionary Computation (ICEC'96), Nagoya, Piscataway NJ 1996.

[19] Bäck T., Hoffmeister F., and Schwefel H.-P., "A survey of evolution strategies" pp. 2-9, in Proceedings of the 4th International Conference on Genetic Algorithms (ICCGA IV), ed. Belewand R. K., and Booker L.B., Mongan Koufman Publishers, Inc., San Diego, 1991.

[20] NEOS server for optimisation, URL http://www-neos.mcs.anl.gov,

[21] Box, George E.P. and Draper, N.R., *Empirical model-building and response surface*, New York, Wiley, 1987.

[22] Sawyer's Gas Turbine Engineering Handbook, Volume I, Theory and Design, Gas Turbine Publications, Inc., 1976.

[23] Durelli A.J., Dally J.W. and Riley W.F., "Stress and strength studies on turbine blade attachments" Proc. SESA 16(1), 171-182.

[24] Gontarovskii P.P. and Kirkach B.N., "Investigation of the stress-strain state of turbine blade root attachments by the finite element method", Strength of Materials, 14: (8) 1037-1041, 1982, Translated from Problemy Prochnosti, No. 8, pp. 37-40, August, 1982.

[25] Lee R.L. and Loh D.L., "Structural Optimisation of Turbine Blade Firtrees", AIAA-88-2995, AIAA/ASME/SAE/ASEE 24th Joint Propulsion conference, July 11-13, 1988, Boston, Massachusetts.

[26] Ibrahim Z., *CAD/CAM Theory and Practice,* McGraw-Hill, Inc., 1991.

[27] Weston, R. P., Townsend, J.C., Eidson, T.M. and Gates, R.L. "A Distributed Computing Environment for Multidisciplinary Design", 5th AIAA/USAF/NASA/ ISSMO Symposium on Multidisciplinary Analysis and Optimisation, Panama City, FL, 1994. (Published in AIAA CP 9413, pp. 1091-1097, also available at NASA Langley Technical Report Server URLhttp://techreports.larc.nasa.gov/ltrs/1994.html.

[28] Townsend, J.C., Samarah, J.A., Weston, R.P. and Zorumski W.E., Integration of a CAD system into an MDO Framework, NASA/TM-1998-207672.

[29] Piegl, L. and Tiller, W., *The NURBS Book*, second edition, Springer-Verlag, 1997.

[30] Pro/Engineer, http://www.ptc.com, 2002.

[31] Schramm, U. and Pilkey, W.D., "The coupling of geometric description and finite elements using NURBS – A study in shape optimisation", Finite elements in analysis and design 15 (1993) 11-34.

[32] Chang, K.H. and Choi, K.K., "A Geometry-based parameterisation method for shape design of elastic solids", Mechanical Structures and MACH., 20(2), 215-252, 1992.

[33] Holzleriner L. and Mahmoud K. G., "Structural shape optimisation using MSC/NASTRAN and sequential quadratic programming", Computers and Structures, 70(1999), 487-514.

[34] Braibant, V. and Fleury, C., "Shape Optimal Design Using B-Splines", Computer methods in applied mechanics and engineering, 44, 247-267, 1984.

[35] PATRAN, http://www.mscsoftware.com/support/prod_support/patran/.

[36] Waldman, W., Heller, M. and Chen, G. X. "Optimal free-form shapes for shoulder fillets in flat plates under tension and bending", International Journal of Fatigue, 23, pp509-523, 2001.

[37] Schramm U., Piley W.D., Deviries R.I., and Zebrowski M.P., "Shape design for thin-walled beam cross sections using rational B Splines", AIAA Journal, Vo.l.33, No.11, November 1995.

[38] Lépine, J., Guibault, F., Trépanier, J.-Y. and Pépin, F., "Optimized Nonuniform Rational B-Spline Geometrical Representation for Aerodynamic Design of Wings", AIAA Journal, Vol. 39, No. 11, November 2001.

[39] Song, W., Keane, A.J. Rees, J., Bhaskar, A. and Bagnall, S., "Local Shape Optimisation of Turbine Disc Firtrees using NURBS", AIAA-2002-5486, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation", Atlanta, Georgia, 4-6 Sep 2002.

[40] Shpitalni, M. and Lipson, H., 1997, "Automatic reasoning for design under geometric constraints," *Annals of the CIRP*, Vol. 46/1, pp. 85-89.

[41] Guy L. Steele Jr, *Common Lisp: the language*, Second Edition, Butterworth-Heinemann, 1990.

[42] Bagnall S.M., "Turbine Blade Root and Disc Head Design – 2D Geometric Parameters and Design Constraints", Rolls-Royce plc. Technical Report, 2000.

[43] Balabanov, V.O., Giunta, A.A., Golovidov, O., Grossman, B., Mason, M.H., Watson, L.T., and Haftka R. T., "Reasonable Design Space Approach to Response Surface Approximation", Journal of Aircraft, Vol. 36, No. 1, January – February, 1999.

[44] Boddington P.H., Chen K. and Ruiz C., "The numerical analysis of dovetail joints", Computers and Structures, Vol. 20, No. 4, pp. 731-735, 1985

[45] Dibsky P., Sergeev B. and Shkarayev S., "Study of the stressed state of dovetail joints", Translation of the article originally printed in Russian in: "Strength of Aircraft Structures," Kharkov, pp. 30-39. 1990.

[46] Papanikos P., Maguid S.A., Stjepanovic Z., "Three-dimensional nonlinear finite element analysis of dovetail joints in aeroengine discs", finite elements in analysis and design 29(1998) 173-185.

[47] Sinclair G.B., Cormier N.G., Griffin J.H., and Meda G., "Contact Stresses in Dovetail Attachments: finite element modelling", Transactions of the ASME, Vol.124, January 2002, pp 182-189.

[48] Sinclair G.B., and Cormier N.G., "Contact Stresses in Dovetail Attachments: Physical Modelling", Journal of Engineering for Gas Turbines and Power, Vol. 124, April 2002, pp 325-331.

[49] Waterhouse R.B. (Ed.), *Fretting Fatigue*. Applied Science, New York, 1981.

[50] Cazenove O., "T900X-6B-HP Turbine Blade Low Number Off Optimisation", 1999. Rolls-Royce plc.

[51] Eldred M.S., "Optimisation strategies for complex engineering applications", Sandia Report, SAND98-0340, February 1998.

[52] Schwefel H.P., 1995, *Evolution and Optimum Seeking*, John Wiley & Sons. 1995.

[53] Lawrence C.T. and Tits A.L., "A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm", SIAM J. Optimisation, Vol. 11, No. 4, pp 1092-1018, 2001.

[54] Kawarada, H., Ohtomo, T., Periaux, J. and Suito, H., "Multi-Start Fuzzy Optimisation Method", GAKUTO International Series, Mathematical Sciences and Applications, Vol. 11, Recent Developments in Domain Decomposition Methods and Flow Problems, 1997.

[55] Boese, K.D., Kahng, A.B., and Muddu, "A new adaptive multi-start technique for combinatorial global optimisations", Operations Research Letters 16 (1994), pp 101-113.

[56] Bonachea, D., Ingerman, E., Levy, J., and McPeak, S., "An improved Adaptive Multi-Start Approach to Finding Near-Optimal Solutions to the Euclidean TSP", Genetic and Evolutionary Computation Conference (GECCO-2000), Las Vegas, Nevada, July 2000.

[57] Eldred, M.S., Giunta, A.A., van Bloemen Waanders, B.G., Wojtkiewicz, S.F., Hart, W.E., and Alleva, M.P., "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimisation, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 3.0 Users Manual", Sandia Technical Report SANDA2001-3796, April 2002.

[58] Sheng L. and Kapania, R.K., "Genetic Algorithms for Optimization of Piezoelectric Actuator Locations", AIAA Journal, Vol. 39. No.9, pp 1818-1822, 2001.

[59] Nair P.B. and Keane A.J., "Passive Vibration Suppression of Flexible Space Structures via Geometric Redesign", AIAA Journal, Vol. 39 No.7, pp 1338-1346, 2001.

[60] Vicini A. and Quagliarella D., "Airfoil and wing design using hybrid optimization strategies", AIAA Journal, Vol.37, No.5, 1999.

[61] Bersini H. and Renders B., "Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways". IEEE International Symposium Evolutionary Computation, pages 312–317, Orlando, Fl., 1994.

[62] Knowles J.D., "Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimisation", PhD thesis, Department of Computer Science, the University of Reading, 2002.

[63] Anderberg, M.R., *Cluster Analysis for Applications,* Academic Press, 1973.

[64] Yin X. and Germay, N., "A Fast Genetic Algorithm with Sharing Scheme Using Cluster Methods in Multimodal Function Optimization", Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, ed. Albrecht, R.F., Reeves, C.R., and Steele, N.C., Springer-Verlag, Innsbruck, pp450-457, 1993.

[65] Kowalik, J., and Osborne, M.R., *Methods for Unconstrained Optimization Problems,* Elsevier, New York, 1968.

[66] Schmit L.A., and Farshi, B., "Some Approximation Concepts for Structural Synthesis", AIAA Journal, Vol. 12, No. 5, 1974, pp 692-699.

[67] Vanderplaats, G. N., and Salajegheh, E., "A New Approximation Method for Stress Constraints in Structural Synthesis", AIAA Journal, Vol. 27, No. 3, 1989, pp 352-358.

[68] Canfield, R. A., "High-Quality Approximations of Eigenvalues in Structural Optimisation", AIAA Journal, Vol. 28, No. 6, 1990, pp.1116-1122.

[69] Hansen, S.R., and Vanderplaats, G. N., "An Approximation Method for Configuration Optimisation of Trusses", AIAA Journal, Vol. 28, No. 1, 1990, pp.161-172.

[70] Ahn, J. A., Kim H., Lee D., and Rho O., "Response Surface Method for Airfoil Design in Transonic Flow", Journal of Aircraft, Vol. 38, No. 2, March-April 2001.

[71] Venter G., Haftka R.T., and Starners J. H. Jr., "Construction of Response Surface Approximations for Design Optimisation", AIAA Journal, Vol.36, No.12, December, 1998

[72] Simpson T.W., "Comparison of response surface and kriging models in the multidisciplinary design of an aerospike nozzle", NASA/CR-1998-206935, ICASE report No. 98-16, February, 1998

[73] Donald R. Jones, Mathias Schinlau and William J. Welch, "Efficient Global optimization of expensive black-box functions", Journal of Global Optimization 13: 455-492, 1998

[74] Bishop, C., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[75] Sacks, J., William J. Welh, Joby J. Mitchell and Henry P. Wynn, "Design and analysis of computer experiments", Statistical Science, Vol. 4, No. 4, pp409-435, 1989.

[76] Dennis, J.E. Jr. "Surrogate Modeling and space mapping for engineering optimization: A summary of the Danish Technical University", November 2000 Workshop.

[77] Jongsoo Lee and Prabhat Hajela, "Application of classifier systems in improving response surface based approximations for design optimization", Computers and Structures 79(2001) 333-344

[78] Guinta A.A. and Watson L.T., "A comparison of approximation modelling techniques: polynomial versus interpolating models", AIAA-98-4758.

[79] Daberkow D.D. and Marris D.N., "New Approaches to conceptual and preliminary aircraft design: A comparative assessment of a Neural network formulation and a response surface methodology", AIAA, 1998 World Aviation Conference, September 28-30, 1998, Anaheim, CA.

[80] Jin R., Chen W., and Simpson T. W., "Comparative Studies of Metamodelling Techniques under Multiple Modelling Criteria", AIAA-2000-4801.

[81] Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates", *Structural Optimization*, to appear, subject to suitable revision.

[82] Alexandra, N., Dennis J.E., Lewis R.M., Virginia Torczon, "A trust region framework for managing the use of approximation models in optimization", NASA/CR-201745, October 1997.

[83] Alexandra, N., Lewis R.M., "First-order frameworks for managing models in engineering optimization", 1st International Workshop on surrogate modelling and space mapping for Engineering. Optimisation, 11/16-19/2000, TDU.

[84] Alexandra, N., "On managing the use of surrogates in general nonlinear optimization and MDO", AIAA-98-4798, 1998.

[85] Byrd R.H., Gilbert J.C. and Nocedal J., "A trust region method based on interior point techniques for nonlinear programming", Institute National de Recherche en informatique et en antomatique.

[86] Giunta, A.A. and Eldred M.S., "Implementation of A trust region model management strategy in the Dakota optimisation toolkit", AIAA-2000-4935

[87] Sellar, R.S., Batill S.M. and Renaud J. E., "Response Surface Based, Concurrent Subspace Optimization for Multidisciplinary System Design"

[88] Wujek, B.A. and J.E. Renaud, "New adaptive Move-limit Management Strategy for approximate optimization, Part 1", AIAA Journal Vol. 36, No.10, October 1998, pp1911-1921

[89] Wujek, B.A. and J.E. Renaud, "New adaptive Move-limit Management Strategy for approximate optimization, Part 2", AIAA Journal Vol. 36, No.10, October 1998, pp1922-1934

[90] Keane A. J., and Petruzzelli N., "Aircraft Wing Design Using GA-Based Mulit-Level Strategies", AIAA-2000-4937, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation", 6-8 Sept. 2000, Long Beach, CA.

[91] Nair, P. B. and Keane A.J., "Combining approximation concepts with genetic algorithm-based structure optimisation procedure", 1998

[92] Ratle, 1998, "Accelerating the convergence of evolutionary algorithms by fitness landscape approximation," Parallel Problem Solving from Nature V, pp. 87-96, 1998.

[93] El-Beltagy M. A. and Keane A.J., "Evolutionary Optimisation for Computationally expensive problems using Gaussian Processes", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99), Morgan Kaufman, pp. 196-203., 1999.

[94] Liang K.H., Yao X., and Newton C., "Evolutionary search of approximated N-dimensional landscapes", International Journal of Knowledge-Based Intelligent Engineering Systems, Vol. 4, No. 3, pp.172-183., 2000.

[95] Jin Y., Olhofer M., and Sendhoff B., "A Framework for Evolutionary Optimization with Approximate Fitness Functions". IEEE Transactions on Evolutionary Computation, 2002. (accpeted).

[96] Simpson T. W., "A Concept Exploration Method for Product Family Design", PhD thesis, Georgia Institute of Technology, 1998.

[97] Morris, M. D., Mitchell, T. J. and Ylvisaker, D., "Baysian design and analysis of computer experiments: use of derivatives in surface prediction", Technometrics, Vol. 35, pp 243-255. 1993.

[98] Bischof, C., Carle, A., Khademi, P., and Mauer, A., "ADIFOR 2.0 : Automatic Differentiation of Fortran 77 Programs", IEEE Computational Science and Engineering, Vol. 3, No. 3, 1996, pp 18-32.

[99] Matthew, GAlib A C++ Library of Genetic Algorithm Components. URL: http://www.lancet.mit.edu/ga/.

[100] Keane A. J., "A Brief Comparison of Some Evolutionary Optimisation Methods" pp. 255-272 in Modern Heuristic Search Methods, ed. V. Rayward-Smith, I. Osman, C. Reeves and G. D. Smith,J. Wiley (ISBN 0 471 96280 5) (1996).

Appendix A Structure of ICAD LISP Geometry

Introduction

The rule-base geometry modelling is implemented in the ICAD system from Knowledge Technologies International (KTI), which is a knowledge-based engineering system with geometric modelling capabilities. The IDL language used in ICAD is a subset of Common Lisp with geometry modelling expansions. The geometry is modelled using object-oriented concepts and consists of a number of building blocks, which again are composed of smaller building blocks, until the basic geometric entities provided by the system itself; thus a hierarchic structure is formed. Several other routines exists providing auxiliary functions such as data I/O, some mathematical computations and some advanced geometric functions. Version control facilities provided by ICAD were utilized to help maintenance and upgrade of the system due to various reasons such as the change of interface to other packages, or the inclusion of additional properties.

Hierarchy of the turbine blade firtree geometry

Each firtree/disc assembly consists of one or three firtree/disc profiles, corresponding to one-sector and three sectors, respectively. Every firtree profile consists of the multi-teeth profile and other connecting geometric entities. Every disc profile also contains the multi-teeth profile and connecting entities. Multiple teeth are generated by duplicating the basic tooth profile, possibly using different sets of parameters, therefore allowing the different teeth size in the model. The hierarchy structure and relations between the components are shown in Figure II.1.

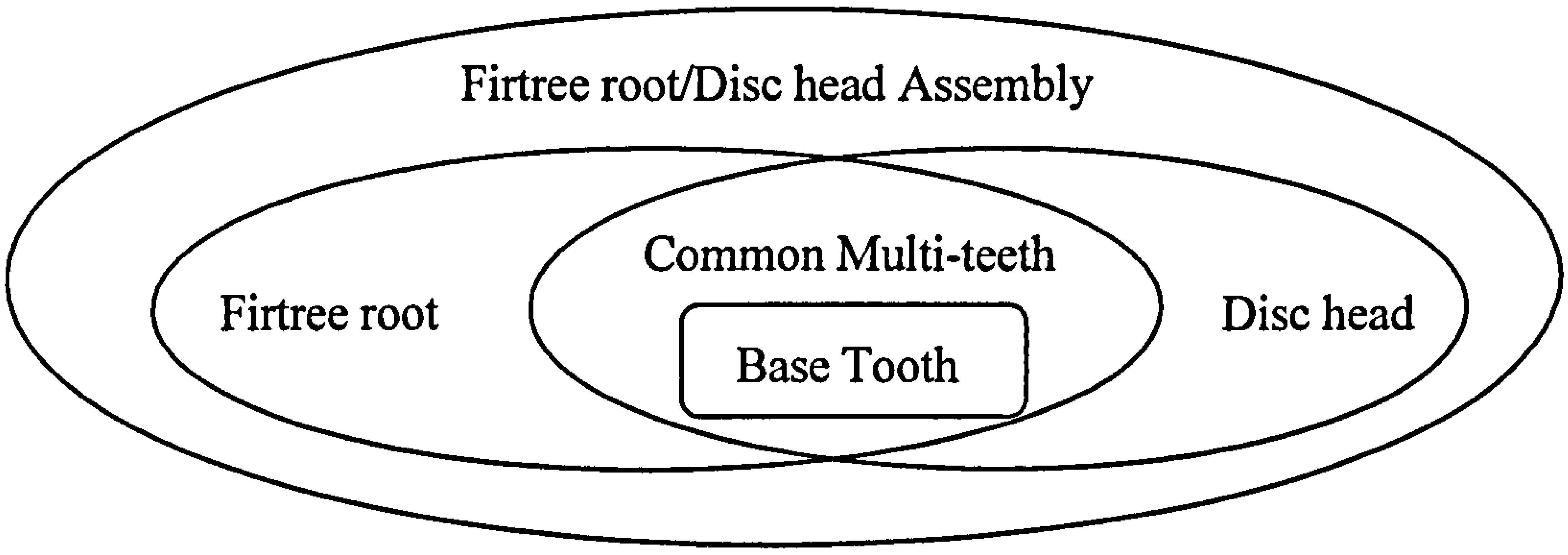


Figure B.1: Illustration of the relations between firtree geometric entities

IDL definition of the base tooth geometry (extract)

This section contains an extract of the IDL code for the base tooth definition. Geometric entities shown in Figure 2.6 are included in the code to illustrate the relationships between these entities.

```
;; An extract of IDL code for base tooth definition
:properties
(
;;( other properties removed )
:curve-list (list (the :disc-crest)
                  (the :blade-trough)
                  (the :blade-crest)
                  (the :disc-trough)
                  (the :disc-line)
                  (the :blade-line)
                  (the :forward)
                  (the :backward)
                  (the :bedding-line)
                  (the :next-bedding-line))
)
:pseudo-parts
(
;; (only single-arc tooth profile related entities retained )

(next-bedding-line :type linear-curve
   :suppress-warning-about (:cad-name)
   :line-constraints
   (:through-point (the :end-point)
    :at-angle (:angle (the :next-top-flank-angle)
    :reference-vector (the (:face-normal-vector :right))
    :plane-normal (the (:face-normal-vector :bottom)))))

(disk-crest :type arc-curve
   :suppress-warning-about (:cad-name)
```

```
   :cad-name (format nil "~adc~d" (the :prefix) (+ (the :index) 1))
   :end-angle (the-child (:angle-at-point (the :disk-line :start)))
   :arc-constraints
   (:center (the :disk-crest-center)
    :radius (the :disk-crest-radius)
    :start-vector (reverse-vector (the :left-arc-ref-vector))
    :plane-normal (the (:face-normal-vector :bottom))))

(blade-trough :type arc-curve
   :suppress-warning-about (:cad-name)
   :cad-name (format nil "~aft~d" (the :prefix) (+ (the :index) 1))
   :end-angle (the-child (:angle-at-point (the :blade-line :start)))
   :arc-constraints
   (:center (the :blade-trough-center)
    :radius (the :blade-trough-radius)
    :start-vector (reverse-vector (the :left-arc-ref-vector))
    :plane-normal (the (:face-normal-vector :bottom))))

(bedding-line :type linear-curve
    :suppress-warning-about (:cad-name)
    :cad-name (format nil "~abl~d" (the :prefix) (+ (the :index) 1))
    :line-constraints
    (:through-point (the :start-point)
     :at-angle(:angle (the :top-flank-angle)
     :reference-vector (the (:face-normal-vector :right))
     :plane-normal (the (:face-normal-vector :bottom)))
     :trim-start   (the :blade-crest (:tangent-point 0))
     :trim-end (the :start-point)))

(disk-trough :type arc-curve
    :suppress-warning-about (:cad-name)
    :cad-name (format nil "~adt~d" (the :prefix) (+ (the :index) 1))
    :fillet
    (:tangent-to
     (:line (the :bedding-line)
      :side-vector (the (:face-normal-vector :front)))
     :tangent-to
     (:line (the :disk-line)
     :side-vector (the (:face-normal-vector :rear)))
     :radius (the :disk-trough-radius)))

(blade-crest :type arc-curve
     :suppress-warning-about (:cad-name)
     :cad-name (format nil "~afc~d" (the :prefix) (+ (the :index) ))
     :fillet
     (:tangent-to
      (:line (the :bedding-line)
       :side-vector (the (:face-normal-vector :front)))
     :tangent-to
     (:line (the :blade-line)
      :side-vector (the (:face-normal-vector :rear)))
      :radius (the :blade-crest-radius)))

(disk-line   :type linear-curve
    :suppress-warning-about (:cad-name)
    :cad-name (format nil "~adl~d" (the :prefix) (+ (the :index) 1))
    :line-constraints
    (:tangent-to
     (:arc (the :disk-crest)
     :side-vector (the (:face-normal-vector :rear)))
     :at-angle (:angle (- (the :under-flank-angle))
```

```
     :reference-vector (the (:face-normal-vector :left)))
     :trim-end (the :disk-trough (:tangent-point 1))))

(blade-line :type linear-curve
   :suppress-warning-about (:cad-name)
   :cad-name (format nil "~afl~d" (the :prefix) (+ (the :index) 1))
   :line-constraints
   (:tangent-to
    (:arc (the :blade-trough)
     :side-vector (the (:face-normal-vector :rear)))
    :at-angle (:angle (- (the :under-flank-angle))
    :reference-vector (the (:face-normal-vector :left)))
    :trim-end (the :blade-crest (:tangent-point 1))))

(backward :type (if (near-to?
            (the :backward-length) 0 :tolerance   0.000001)
            'null-part 'linear-curve)
 :suppress-warning-about (:cad-name)
 :cad-name  (format  nil  "~abw~d"  (the  :prefix)   (+  (the  :index)  1))
:line-constraints
 (:through-point (the :end-point)
  :at-angle(:angle (the :next-top-flank-angle)
  :reference-vector (the (:face-normal-vector :left))
  :plane-normal (the (:face-normal-vector :bottom))
  :length (the :backward-length))))

(forward :type (if (near-to?
            (the :forward-length) 0 :tolerance 0.000001)
            'null-part 'linear-curve)
 :suppress-warning-about (:cad-name)
 :cad-name (format nil "~afw~d" (the :prefix) (+ (the :index) 1))
 :line-constraints
 (:through-point (the :blade-crest (:tangent-point 0))
  :through-point (the :disk-trough (:tangent-point 0))))
)

:parts
((tooth :type composed-curves
        :suppress-warning-about (:cad-name)
        :curves (the :curve-list)
        :suppress-internal-gaps? t
        :closed? nil)

)
```

Lisp command files used to generate the geometry and related data files

A lisp command file is executed by the ICAD system to produce the required geometry and related data files. Details of the lisp command file follow:

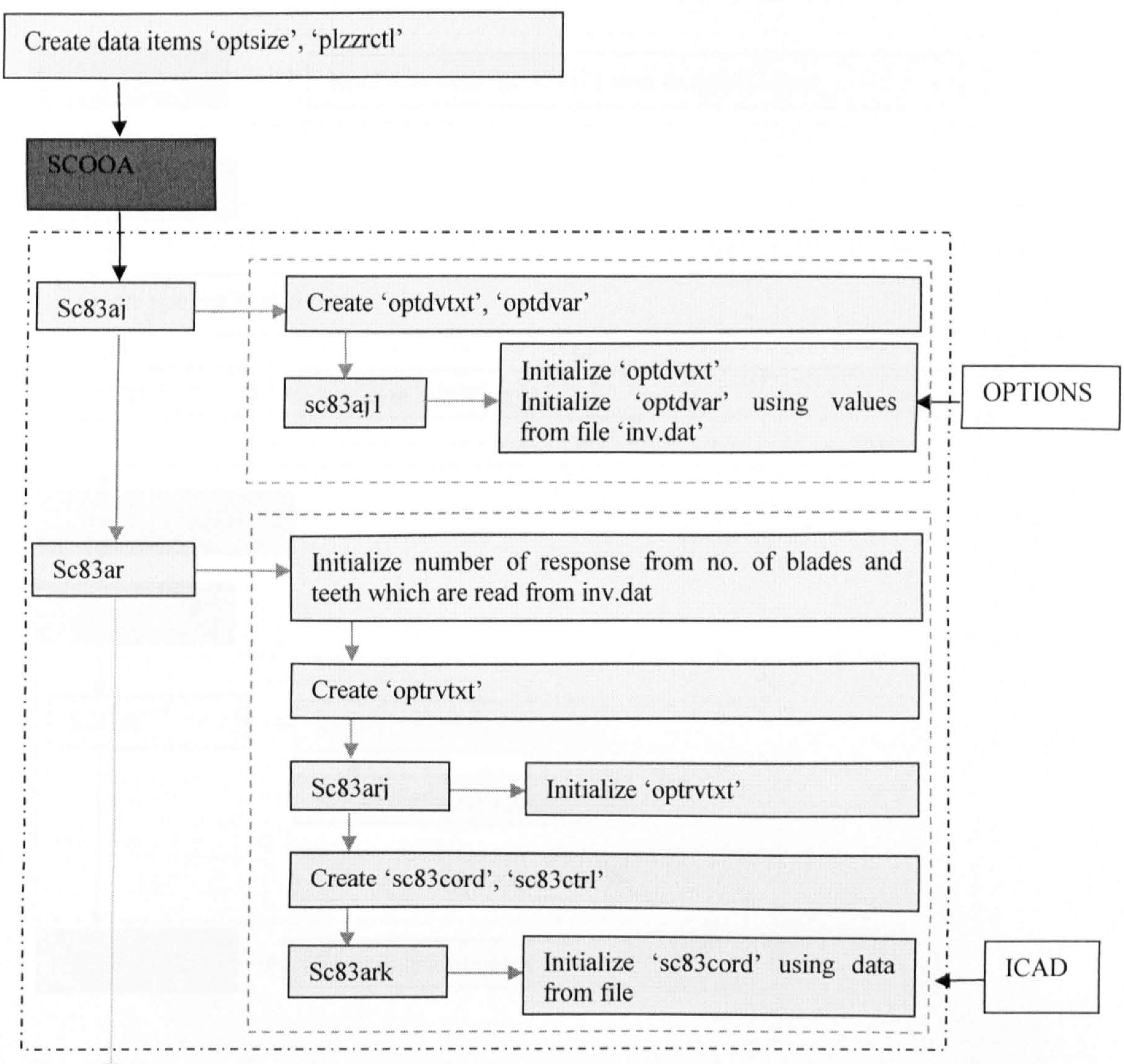
```
(in-package :idl-user)
(load "/home1/utp-11/sow/pub/kbe/registry/kbe.host")
(add-registry-directory "kbe:registry;")
(defparameter *project-source-directory*
```

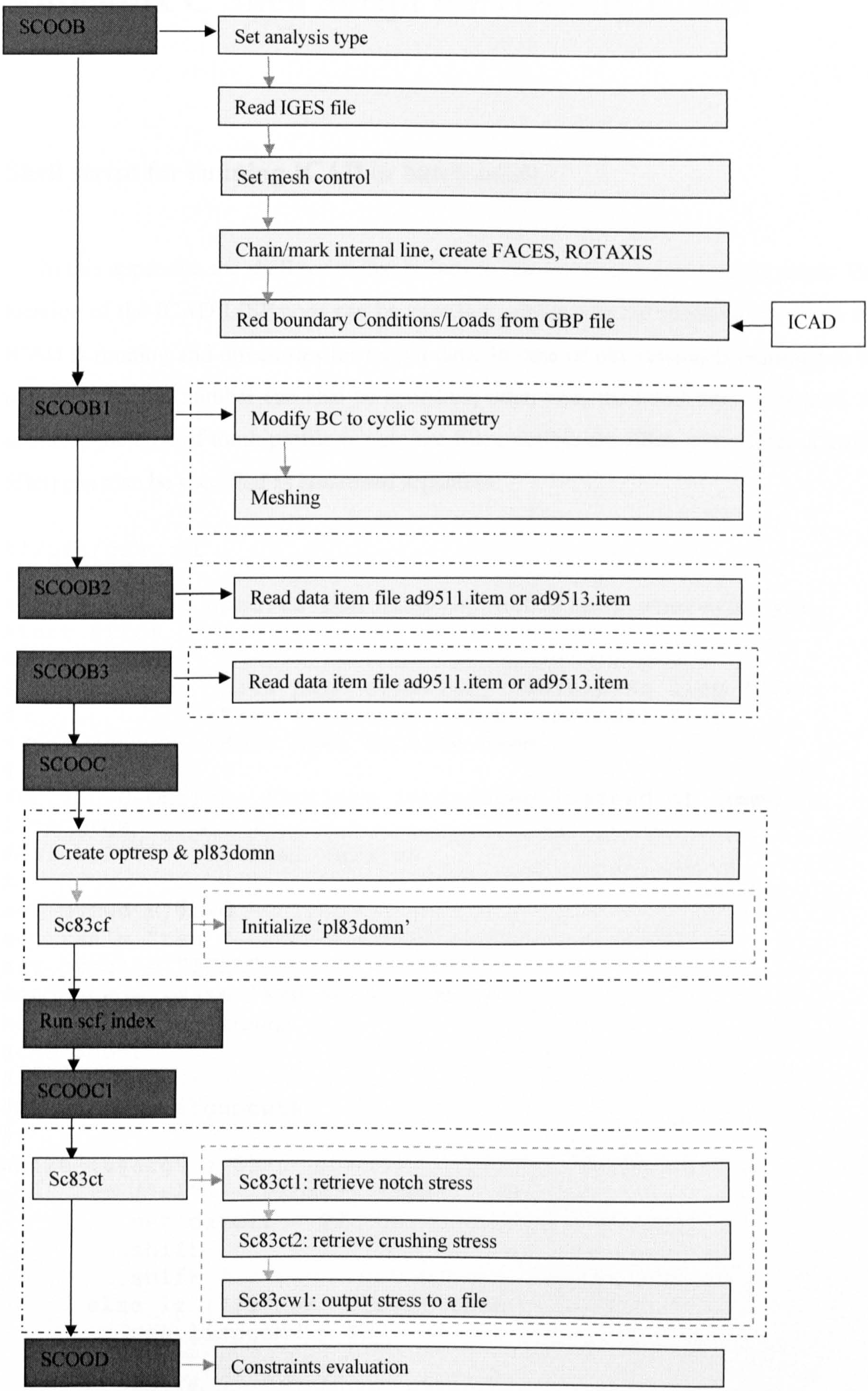
```
    ;; Dont forget the trailing slash
"/home1/utp-11/sow/firtree/trent/")
(defparameter *variable-file*
            "/home2/utp-11/sow/tmp/sow-1959/inv.dat")
(ignore-errors (make-solid-modeler :parasolid :in-memory))
(make-solid-modeler :parasolid :in-memory)
(defparameter *parameter-file*
            "/home1/utp-11/sow/firtree/drf/ind10.dat")
(defparameter *working-directory*
            "/home2/utp-11/sow/tmp/sow-1959/")
(defparameter *obj-file*
            "/home2/utp-11/sow/tmp/sow-1959/obj.dat")
(defparameter *con-file*
            "/home2/utp-11/sow/tmp/sow-1959/geo.dat")
(defparameter *iges-file*
            "/home2/utp-11/sow/tmp/sow-1959/firone2.iges")
(defparameter *gbp-file*
            "/home2/utp-11/sow/tmp/sow-1959/firone2.gbp")
(defparameter *geo-file*
            "/home2/utp-11/sow/tmp/sow-1959/firone2.geo")
(defparameter *sc03-item-file*
            "/home2/utp-11/sow/tmp/sow-1959/mesh.item")
(defparameter *status-file-name*
            "/home2/utp-11/sow/tmp/sow-1959/status.dat")
(load-system :firtree :version 56)
(setf *firtree-disk-1* (make-part 'firtree-disk))
(the-object *firtree-disk-1* :store-status)
(the-object *firtree-disk-1* :iges-output)
(the-object *firtree-disk-1* :geo-response)
(the-object *firtree-disk-1* :gbp-output)
(the-object *firtree-disk-1* :store-objective)
(the-object *firtree-disk-1* :store-constraints)
(the-object *firtree-disk-1* :notch-cord)
(the-object *firtree-disk-1* :sc03-item)
(defparameter *parameter-file*
              "/home1/utp-11/sow/firtree/drf/ind30.dat")
(defparameter *iges-file*
              "/home2/utp-11/sow/tmp/sow-1959/firthr2.iges")
(defparameter *gbp-file*
              "/home2/utp-11/sow/tmp/sow-1959/firthr2.gbp")
(defparameter *geo-file*
              "/home2/utp-11/sow/tmp/sow-1959/firthr2.geo")
(setf *firtree-disk-3* (make-part 'firtree-disk))
(the-object *firtree-disk-3* :iges-output)
(the-object *firtree-disk-3* :gbp-output)
(the-object *firtree-disk-3* :geo-response)
(the-object *firtree-disk-1* :store-objective)
(the-object *firtree-disk-1* :store-constraints)
```

Appendix B Development of SC03 Plugin

Introduction

The finite element package SC03 provides a mechanism to expand its capabilities on top of its core analysis functionalities. Such implementations are termed plugins. In this work, an optimisation interface plugin was developed to carry out the automatic finite element analysis of the turbine blade firtrees. Tasks include the transfer of geometry through the IGES standard; application of boundary conditions and loads; specification of material properties and thickness values, etc. Details on the program structure are given below.




SCOOB
Set analysis type
Read IGES file
Set mesh control
Chain/mark internal line, create FACES, ROTAXIS
Red boundary Conditions/Loads from GBP file
ICAD
SCOOB1
Modify BC to cyclic symmetry
Meshing
SCOOB2
Read data item file ad9511.item or ad9513.item
SCOOB3
Read data item file ad9511.item or ad9513.item
SCOOC
Create optresp & pl83domn
Sc83cf
Initialize 'pl83domn'
Run scf, index
SCOOC1
Sc83ct
Sc83ct1: retrieve notch stress
Sc83ct2: retrieve crushing stress
Sc83cw1: output stress to a file
SCOOD
Constraints evaluation

Appendix C Shell Script for running ICAD

Shell script for running ICAD in batch mode

In this appendix, the shell script that is used to run ICAD in batch mode is given. The location of the ICAD LISP code can be specified, along with the machine on which the ICAD is running and directories for output data. In case of old version is expected to be used, the version number can also be specified; otherwise, the latest version is used. In addition, the type of tooth profile (single-arc fillet, double-arc fillet, conic fillet or cubic fillet) can also be specified as command argument.

```
#!/bin/csh  -f
#
#Shell script used to run ICAD on achilles, wherever you
start #from.
#
#Options are: -Data path directory containing lisp files
#             -Exec file exec (lisp commands) file
#             -Host host machine name
#
#Oct.2000-Options facility introduced instead of command
line args
#March,2000 - Initial version
#
set mycwd = $cwd
set rsh = "rsh "
set progl = "ICAD"
set date = `date '+%d-%b-%y  %H:%M'`
set host = `hostname`
echo $host
#
### Process arguments
#
while ($#argv > 0)
      if ("$1" =~ -[Dd]*) then
         set datdir = $2
         shift
         shift
      else if ("$1" =~ -[Ee]*) then
         set lispcomd = $2
         shift
         shift
```

```
      else if ("$1" =~ -[Tt]*) then
         set toothtype = $2
         shift
         shift
      else if ("$1" =~ -[Vv]*) then
         set lispversion = $2
         shift
         shift
      else if ("$1" =~ -[Ww]*) then
         set tmpdir = $2
         shift
         shift
      else if ("$1" =~ -[Hh]*) then
         set machine = $2
         if($machine == "") then
              echo "C-Shell Script for running ICAD system"
              echo "Options are:"
              echo "   -Directory            data item
directory"
              echo "   -Work                 Work directory"
              echo "   -Exec  file     exec (lisp command)
file"
              echo "   -Host      host       machine name"
              exit(0)
          endif
          shift
          shift
      else
          set inp = $1
          shift
      endif
end
#
# get defaults
#
if(! $?datdir) then
    set datdir=$mycwd
endif
if(! $?tmpdir) then
    set tmpdir=$mycwd
endif
set mchdef="achilles"
if(! $?machine) then
  set machine=$mchdef
endif
if (-e $HOME/.icadsgdef) then
  set mchdef = `cut -d' ' -f1 $HOME/.icadsgdef`
endif
echo $datdir
echo $tmpdir
echo $mchdef
```

```
#
# save defaults
#
echo $machine >$HOME/.icadsgdef
#
# $: icad model directory, currently /homel/utp-
11/sow/icad/fd
# $: working directory, /usr/tmp/sow-pid
# $: objective function
# $: geometric constraints

#cat > $tmpdir/icad-lisp.commands << eof

; Work versions of code/
echo '(in-package :idl-user)'  >$tmpdir/icad-lisp.commands

;; define the ICAD model
echo '(load "/homel/utp-11/sow/pub/kbe/registry/kbe.host")'
>> $tmpdir/icad-lisp.commands
echo '(add-registry-directory "kbe:registry;")'>>
$tmpdir/icad-lisp.commands
echo '(defparameter *project-source-
directory*'>>$tmpdir/icad-lisp.commands
echo '    ;; Dont forget the trailing slash'
>>$tmpdir/icad-lisp.commands
echo '"'$datdir'/")'  >>$tmpdir/icad-lisp.commands

# use the concept of system, so a defintion of *firtree-
model* is #no longer required
#echo '(defparameter *firtree-model*
"'$datdir'/project.lisp")' >>$tmpdir/icad-lisp.commands
echo '(defparameter *variable-file*   "'$tmpdir'/inv.dat")'
>>$tmpdir/icad-lisp.commands

echo '(ignore-errors (make-solid-modeler :parasolid :in-
memory))'  >>$tmpdir/icad-lisp.commands
echo '(make-solid-modeler :parasolid :in-memory)'
>>$tmpdir/icad-lisp.commands

;; One sector model
# parameter file for different type of tooth
if($toothtype == "srf") then
echo '(defparameter *parameter-file*   "/homel/utp-
11/sow/firtree/srf/ins10.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "asrf") then
echo '(defparameter *parameter-file*   "/homel/utp-
11/sow/firtree/srf/ins11.dat")'  >>$tmpdir/icad-
lisp.commands
endif
```

```
if($toothtype == "drf") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/drf/ind10.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "adrf") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/drf/ind11.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "conic") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/conic/inc10.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "cubic") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/cubic/inq10.dat")'  >>$tmpdir/icad-
lisp.commands
endif
echo '(defparameter *working-directory* "'$tmpdir'/")'
>>$tmpdir/icad-lisp.commands
echo '(defparameter *obj-file*         "'$tmpdir'/obj.dat")'
>>$tmpdir/icad-lisp.commands
echo '(defparameter *con-file*         "'$tmpdir'/geo.dat")'
>>$tmpdir/icad-lisp.commands
echo '(defparameter *iges-file*
"'$tmpdir'/firone2.iges")'  >>$tmpdir/icad-lisp.commands
echo '(defparameter *gbp-file*
"'$tmpdir'/firone2.gbp")'  >>$tmpdir/icad-lisp.commands
echo '(defparameter *geo-file*
"'$tmpdir'/firone2.geo")'  >>$tmpdir/icad-lisp.commands
echo '(defparameter *sc03-item-file*
"'$tmpdir'/mesh.item")'  >>$tmpdir/icad-lisp.commands
echo '(defparameter *status-file-name*
"'$tmpdir'/status.dat")'  >>$tmpdir/icad-lisp.commands

#echo '(load *firtree-model*)'  >>$tmpdir/icad-lisp.commands
# instead of load a set of user-specified lisp files, a
system is loaded, this will make sure that
# the latest version or deployed version is used
echo '(load-system :firtree :version '$lispversion')' >>
$tmpdir/icad-lisp.commands
echo "(setf *firtree-disk-1* (make-part 'firtree-disk))"
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :store-status)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :iges-output)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :geo-response)'
>>$tmpdir/icad-lisp.commands
```

```
echo '(the-object *firtree-disk-1* :gbp-output)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :store-objective)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :store-constraints)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :notch-cord)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :sc03-item)'
>>$tmpdir/icad-lisp.commands


;; three sectors model
# parameter file for different type of tooth
if ($toothtype == "srf") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/srf/ins30.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if ($toothtype == "asrf") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/srf/ins31.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "drf") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/drf/ind30.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "adrf") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/drf/ind31.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "conic") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/conic/inc30.dat")'  >>$tmpdir/icad-
lisp.commands
endif
if($toothtype == "cubic") then
echo '(defparameter *parameter-file*   "/home1/utp-
11/sow/firtree/cubic/inq30.dat")'  >>$tmpdir/icad-
lisp.commands
endif

echo '(defparameter *iges-file*
"'$tmpdir'/firthr2.iges")'  >>$tmpdir/icad-lisp.commands
echo '(defparameter *gbp-file*
"'$tmpdir'/firthr2.gbp")'  >>$tmpdir/icad-lisp.commands
echo '(defparameter *geo-file*
"'$tmpdir'/firthr2.geo")'  >>$tmpdir/icad-lisp.commands
```

```
echo "(setf *firtree-disk-3* (make-part 'firtree-disk))"
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-3* :iges-output)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-3* :gbp-output)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-3* :geo-response)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :store-objective)'
>>$tmpdir/icad-lisp.commands
echo '(the-object *firtree-disk-1* :store-constraints)'
>>$tmpdir/icad-lisp.commands


echo lisp-command-file finished
# Create shell script for ICAD batch run
echo '#\!/bin/csh'                                          >
$tmpdir/icad-shell-commands.sh
#use icad image on /home1/utp-11/sow/icad/utp
#echo 'cat icad-lisp.commands | $UTP_ICAD_HOME/icad -i'
>>$tmpdir/icad-shell-commands.sh
echo 'source ~/.env'
>>$tmpdir/icad-shell-commands.sh
echo 'echo $$ > '$tmpdir'/icadpid.dat'
>>$tmpdir/icad-shell-commands.sh
echo 'cat '$tmpdir'/icad-lisp.commands | $ICADHOME/bin/icad
-i'   >>$tmpdir/icad-shell-commands.sh
echo 'rm -f '$tmpdir'/icadpid.dat'
>>$tmpdir/icad-shell-commands.sh
echo 'exit 0'
>>$tmpdir/icad-shell-commands.sh
chmod +x $tmpdir/icad-shell-commands.sh
#
# submit the job to $machine
#
echo $machine
echo $host
if($machine != $host) then
#  echo "--Checking access to $machine..."
#  set iret1 = `$rsh $machine "$rsh $host tty >& /dev/null;
echo "'$status'`
#  set iret2 = $status
#  set iret3 = $iret1[$#iret1]
#  if($iret2 == 1 || $iret3 == 1) then
#    echo "** You do not have access to $machine"
#    echo "** Make sure you have both $machine and $host in
your .rhosts file"
#    exit 1
#  endif
  set rsh = "rsh"
  $rsh $machine $tmpdir/icad-shell-commands.sh
```

```
  exit 0
else if($machine == $host) then
  echo $rsh
  echo $tmpdir
  $tmpdir/icad-shell-commands.sh
endif
```