ARMY RESEARCH LABORATORY

Multi-body Dynamic Contact Analysis Tool for Transmission Design

SBIR Phase II Final Report

by Sandeep Vijayakar, Samir Abad and Rajendra Gunda

ARL-CR-487 April 2003

prepared by
ADVANCED NUMERICAL SOLUTIONS
3956 BROWN PARK DRIVE, SUITE B
HILLIARD OH 43026

under contract
DAAD17-00-C-0073
for patr:
TIMOTHY L. KRANTZ
AMSRL-VT-E
NASA JOHN H. GLENN RESEARCH CENTER, MS 77-10
21000 BROOKPARK ROAD
CLEVELAND, OH 44135

[illegible]

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Public Release	18. NUMBER OF PAGES 194	19. NAME OF RESPONSIBLE PERSON Rike, Jack jrike@dtic.mil EM931
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number DSN

[illegible]

Multi-body Dynamic Contact Analysis Tool for Transmission Design

SBIR Phase II Final Report

Advanced Numerical Solutions
Hilliard OH

April 28, 2003

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Opportunity

The weight of transmissions is the critical constraint in improving the life, performance and efficiency of rotorcraft. Limitations of available design tools make difficult the task of building of lighter and more reliable transmissions.

Rotorcraft and modern automotive transmissions are designed to be very light. This means that the gears and structural components in the transmission can undergo significant deformation at rated load. The AGMA J factor and Lewis parabola methods are gear design and stress estimation methods commonly used in the gearing industry. The applicability of these and similar methods is limited in lightweight gearing applications due to the narrow gear rims and relatively large deformations in the transmission. Furthermore, because of the large interaction between the various gear pair mesh interfaces, analyzing any one-gear pair mesh in isolation can produce erroneous results. These difficulties are compounded by the fact that any realistic analysis of a lightweight transmission would have to include the dynamic forces. Finally, safety margins are much smaller, and the need for accurate design tools is much more critical.

Researchers have been attempting to address these issues for over a decade. The work by Valco [1] is probably the most recent attempt at a computational model for a multi-mesh gear system. He attempted to build a static two-dimensional finite element model of a planetary gear system with a flexible ring gear. Because of the non-linearity introduced by tooth contact, a commercial non-linear finite element package was used. However, there is a mismatch between the capabilities of such general-purpose commercial programs and the special needs of gear contact analysis. Consequently, run times for that model ran into hundreds of hours on a contemporary supercomputer.

1.2 The Challenge

Several special features make a multi-mesh contact analysis so difficult for general-purpose finite element software:

- *Size of the Contact Zone*: The width of the contact zone in typical gearing applications is two orders of magnitude smaller than the dimensions of the gear teeth themselves. In order to resolve the contact conditions with sufficient accuracy, a general purpose non-linear finite element program needs to have a large number of nodes inside the contact zone. In order for such a contact model to run, the fine mesh in the contact zone has to transition into a much coarser mesh over the rest of the gear. The location of the contact zone, however, changes as the gears move. This implies either that the gear finite

element model be re-meshed for each time instant, or that the finite element mesh be highly refined over its entire surface area. Both these alternatives lead to unacceptably high computational costs. Our approach [3] has been to use the finite element models only to compute relative deformation and stresses for points that are away from the contact zones. For points within the contact zone, we use semi-analytical techniques to compute the relative deformations and stresses. The 'near field' semi-analytical solution and the 'far field' finite element solutions are matched at a 'matching surface'. Such a model is significantly more difficult to program on a computer, but once implemented, can provide much better resolution without using a highly refined finite element mesh.

- *Many rigid body degrees of freedom in the system*: In multi-mesh gear systems like planetary transmissions, there are many rigid body type degrees of freedom or mechanisms that are constrained only by the contact conditions. This means that if a non-linear finite element code with 'gap elements' is used, then the incremental stiffness matrices become singular. Most commercial codes cannot proceed when this happens. Some workarounds are commonly used, such as adding imaginary linear and torsional springs to make the system stiffness matrices non-singular. The spring stiffness can be made small, but the accuracy of results computed by such almost singular stiffness matrices is questionable. Our approach has been to attach a reference frame to each individual component, and to carry out the finite element computations for each individual component separately in its own reference frame. As long as each finite element mesh is sufficiently well constrained to its reference frame, the stiffness matrices are well behaved. The free mechanisms in the system can be modeled by allowing the reference frames to move freely. The contact solver used is based on the Revised Simplex Solver. This Solver is commonly used to solve quadratic programming problems. It can take into account any free mechanisms in the system while computing the contact loads. The only disadvantage of this technique is that is very difficult to program. Again, there is a trade-off between programming complexity and program efficiency.

- *Large number of degrees of freedom*: For a system in which the total number of gear teeth is about 200 or more, the total number of finite element degrees of freedom can be extremely large. This is so even with the finite element model refined only as much as is necessary for the far field solution. Figure 7.12 shows a planetary transfer case model in which the total number of finite element degrees of freedom is approximately 1.5 million. Figure 8.8 shows a rear axle differential gear assembly, another very large gear system model with over a million degrees of freedom. CPU time and memory needed to run a contact analysis with such a large degree of freedom would make it impractical. We have resorted to using a hierarchical representation of the system, in which the system is build from many substructures, with each substructure in turn being composed of many substructures. The processes of stiffness decomposition and load vector back-substitution now become very complex, and involve multiple recursive traversals of the substructure hierarchy. But it is now possible to keep CPU requirements down to a reasonable level. Yet again, this is possible by accepting programming complexity in exchange for an increase in speed.

- *System Kinematics*: The nominal position of each individual gear in the system changes with time. The nominal positions of the components are determined by the kinematics of the system. The kinematics of the system affects the nominal sliding velocities and inertial loads. It is very difficult to include this kinematic information into the finite element programs currently available. We have built a special purpose programming language into the software in order to specify in detail the kinematics of each component in the system. Important details such as the kinematic effect of assembly errors, runout and misalignments are easy to apply using this approach.

- *Convergence of conditions at contact interfaces*: Poor convergence of contact conditions is one of the biggest problems caused by using a general non-linear solver to solve a problem with contact constraints. The constraints imposed by the contact between mating surfaces are essentially linear inequality constraints. When a general-purpose non-linear solver is used to solve this problem, convergence is not guaranteed, and if convergence does occur, it is usually very slow. The Revised Simplex solver that we use provides a guarantee of convergence within a predetermined number of iterations. Furthermore, ill-posed contact problems can be detected even before the solution process is started. The solver is specifically designed for the linear inequality type constraints found in contact problems.

1.3 SBIR Phase I

The objective of Phase I of this SBIR project was to establish the technical feasibility of building multi-body gear contact analysis software. The key milestones that were achieved during Phase I were:

- A prototype was built of the multi-mesh contact analysis program that we call Calyx.
- A prototype was built of a user friendly planetary model generation and post-processing program we call Multyx. At the end of Phase I, the program was not full featured enough to build three-dimensional models, or to adequately post-process results, but the feasibility of completing the task was established.
- The frequency response of our computational Calyx planetary system model was validated against simpler lumped parameter models.
- The computational predictions of dynamic response of a simple spur gear system modeled using Calyx were compared with published experimental results.
- A comparison of stress predictions with published experimental results for external and internal spur gears was carried out..

1.4 SBIR Phase II Objective

Phase II of the SBIR project involves the implementation of all the features in the contact analysis software that would be necessary for it to be usable by gear design engineers. The important milestones that we achieved during Phase II are:

- *Test 2D planetary analysis software*: A significant part of the effort was directed towards testing, debugging and validating the computer programs. The two-dimensional planetary analysis software was tested in static and dynamic mode, and the frequency response spectra were compared with lumped parameter models. After in-house testing, the gear labs at the University of Toledo, and at the Ohio State University subjected the software to further testing, helping us build confidence in the dynamic capability of *Calyx*.
- *Build 2D and 3D Data extraction code*: We built *Multyx* into a full featured, user friendly program that can be used to generate many different graphs, reports and displays for post-processing the two-dimensional and three-dimensional data. A new program called *iGlass* was created to allow a user to interactively view and animate a three-dimensional model and its analysis results.
- *Build 2D and 3D Mesh Generation code*: We incorporated into *Multyx* a special purpose mesh generator for two-dimensional models of spur gears, and three-dimensional models of internal and external spur and helical gears..

- *Test alternative computer platform options* We tested *Calyx* and *Multyx* on personal computers running Microsoft Windows and Linux, workstations running Unix, and on SGI super computers at the Ohio Supercomputer Center.
- *Develop the contact system solver*: Many improvements were made to the contact system solver to improve robustness when dealing with badly conditioned, or marginally well conditioned systems of equations. The original contact solver based on the revised simplex solver was replaced with one based on the quadratic programming method.
- *Roller Bearings*: We extended Calyx's capability to include contact analysis of rolling element bearings. This feature was implemented, and we were able to test the program on cylindrical roller bearings, tapered roller bearings, as well as ball bearings.
- *Implement and test a 3D Spiral bevel and hypoid gear mesh generator*: We successfully implemented a full featured mesh generator for spiral bevel and hypoid pinions and gears based on the face-milling process. This has been tested for several different industrial applications. A mesh generator based on the face hobbing process was also developed, and is undergoing tests. We also built and tested mesh generators for straight bevel gears based on the octoid tooth forms. These are commonly used in differential gear sets.
- *Implement and test 3D planetary system model generators*: The *Planetary3D* and *Transmission3D* software packages were created and tested. These two packages together allow a gear analyst to build a model of virtually any gear system in use today. They are capable of modeling arbitrarily complex simple and compound planetary gear systems.
- *Implement capability to import carrier and housing finite element models from widely used commercial software*: We implemented a program to convert NASTRAN bulk data files into *Calyx* models of carriers and housings. Commercial software was first used to convert industrial housing and carrier models from other formats into NASTRAN bulk data files. These files were then converted into carrier and housing models such as those illustrated in Figures 7.12 and 8.8.
- *Implement shell finite elements*: We successfully implemented and tested a shell finite element formulation in *Calyx*. Two housing models involving shell elements were created and tested with the *Planetary3D* and *Transmission3D* packages.

Several important software packages were created as the result of Phase II. *Planetary2D* is a self-contained package for the static and dynamic analysis of simple two-dimensional planetary gear models. It is already being used and tested by end-users in the automotive industry. *Helical3D* is a package capable of analyzing three-dimensional models of spur and helical gear pairs. It is also currently being used in the industry, and has replaced our older computer program *CAPP*. The *HypoidFaceMilled* package for the analysis of face-milled spiral-bevel and hypoid gears has also been released to end users. A new package called *HypoidFaceHobbed* was also developed using the technology created in Phase II, and will be released soon for user trials. It is capable of modeling spiral-bevel and hypoid gears generated by the face-hobbing method. *Planetary3D* is a package for parallel axis full-transmission models, and *Transmission3D* is a package for parallel and non-parallel axis full-transmission models. These have been implemented and tested in-house.

The Phase II of the SBIR project has helped us develop a technology that has already begun to be used in the automotive and aerospace industry. All the major automotive companies in the United States, are now using some of the technology that resulted from this project. Two of the largest aerospace companies in the country are already evaluating the products that resulted from Phase II.

We forsee a large direct and indirect impact of the Phase II project. The prospects of a successful Phase III venture are very promising.

1.5 Additional Documentation

This report provides an overview of the work carried out under the SBIR Phase II project. Additional detailed information is available in the following manuals that were also prepared for this project.

- *Calyx User's Manual:* Information for the user who needs to directly control the contact analysis program *Calyx*.
- *Planetary2D User's Manual:* Instructions for using the two-dimensional planetary analysis software package *Planetary2D*.
- *Planetary2D Validation Manual:* Examples and comparisons for validating the two-dimensional planetary analysis software package *Planetary2D*.
- *Helical3D User's Manual:* Instructions for using the three-dimensional spur and helical gear analysis software package *Helical3D*.
- *Helical3D Validation Manual:* Validation examples for the three-dimensional spur and helical gear analysis software package *Helical3D*.
- *HypoidFaceMilled User's Manual:* Instructions for using the *HypoidFaceMilled* software package for the analysis of spiral-bevel and hypoid gears manufactured by the face-milling process.
- *HypoidFaceMilled Validation Manual:* Validation examples for the *HypoidFaceMilled* software package.
- *Planetary3D User's Manual:* Instructions for using the *Planetary3D* software package for the three-dimensional modeling and analysis of simple and compound planetary gear sets.
- *Planetary3D Validation Manual:* Validation examples for the *Planetary3D* software package.
- *Transmission3D User's Manual:* Instructions for using the *Transmission3D* software package for the three-dimensional modeling and analysis of transmissions.
- *Transmission3D Validation Manual:* Validation examples for the *Transmission3D* software package.
- *Housing Noise Radiation Theory Manual:* Boundary element theory used for the housing noise radiation prediction.
- *Housing Noise Radiation Validation Manual:* Test cases and comparison for the housing noise radiation calculation procedure.

Chapter 2

Technology

2.1 Dynamic Contact Analysis

We use a finite element formulation unique in its combination of detailed contact modeling between the elastic teeth [3] with a combined surface integral/finite element solution [4] to efficiently capture tooth deformations and loads with a relatively coarse mesh. Details are available in the references and a brief description of the surface integral/finite element solution is given in [6]. The contact analysis is briefly described here.

Each of the bodies undergoes large rotations. However, the deviation of the motion of each body from a predetermined (kinematic) trajectory is very small. If the finite element displacement vector $\mathbf{x}_{fi}$ for a particular body i is measured with respect to a reference frame that follows this known trajectory, then it is possible to represent its behavior by a linear system of equations:

$$\mathbf{M}_{ffi}\ddot{\mathbf{x}}_{fi} + \mathbf{C}_{ffi}\dot{\mathbf{x}}_{fi} + \mathbf{K}_{ffi}\mathbf{x}_{fi} = \mathbf{f}_{fi} \tag{2.1}$$

Here $\mathbf{f}_{fi}$ is a vector of external loads, $\mathbf{M}_{ffi}$ is the mass matrix, $\mathbf{C}_{ffi}$ is the damping matrix and $\mathbf{K}_{ffi}$ is the stiffness matrix for the body. The damping matrix is obtained by using Raleigh's damping model:

$$\mathbf{C}_{ffi} = \mu\mathbf{M}_{ffi} + \eta\mathbf{K}_{ffi} \tag{2.2}$$

If the finite element mesh is constrained adequately to its reference frame, then $\mathbf{x}_{fi}$ will contain no rigid body type of degrees of freedom, and $\mathbf{K}_{ffi}$ will be positive definite. $\mathbf{M}_{ffi}$ and $\mathbf{C}_{ffi}$ are always positive definite.

Rigid body degrees of freedom are assigned, not to the finite element mesh, but to the reference frame. A vector $\mathbf{x}_{ri}$ contains the components of the reference frame's unconstrained rigid body degrees of freedom. In general, there may be up to six components in this vector. For a two-dimensional model, there may be only three unconstrained components. Augmenting 2.1 with the vector $\mathbf{x}_{ri}$, and assuming that $\mathbf{x}_{ri}$ is very small, we obtain a linear relationship:

$$\begin{bmatrix} \mathbf{M}_{ffi} & \mathbf{M}_{fri} \\ \mathbf{M}_{rfi} & \mathbf{M}_{rri} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}}_{fi} \\ \ddot{\mathbf{x}}_{ri} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_{ffi} & \mathbf{C}_{fri} \\ \mathbf{C}_{rfi} & \mathbf{C}_{rri} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{x}}_{fi} \\ \dot{\mathbf{x}}_{ri} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{ffi} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{x}_{fi} \\ \mathbf{x}_{ri} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_{fi} \\ \mathbf{f}_{ri} \end{Bmatrix} \tag{2.3}$$

The terms $\mathbf{M}_{rfi}$, $\mathbf{M}_{fri}$, $\mathbf{M}_{rri}$, $\mathbf{C}_{rfi}$, $\mathbf{C}_{fri}$, $\mathbf{C}_{rri}$, are easily computed from the finite element mesh by using energy methods. Effects of any lumped masses or lumped viscous dampers are added into the matrices $\mathbf{M}_{rri}$ and $\mathbf{C}_{rri}$, respectively, at this stage. The next step is to assemble the equations for each body i into a larger system of equations for the entire system:

$$\begin{bmatrix} \mathbf{M}_{ff} & \mathbf{M}_{fr} \\ \mathbf{M}_{rf} & \mathbf{M}_{rr} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}}_f \\ \ddot{\mathbf{x}}_r \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_{ff} & \mathbf{C}_{fr} \\ \mathbf{C}_{rf} & \mathbf{C}_{rr} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{x}}_f \\ \dot{\mathbf{x}}_r \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{ff} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{x}_f \\ \mathbf{x}_r \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_f \\ \mathbf{f}_r \end{Bmatrix} \quad (2.4)$$

where

$$\mathbf{x}_f = \begin{Bmatrix} \mathbf{x}_{f1} \\ \mathbf{x}_{f2} \\ \vdots \\ \mathbf{x}_{fi} \\ \vdots \end{Bmatrix}, \mathbf{x}_r = \begin{Bmatrix} \mathbf{x}_{r1} \\ \mathbf{x}_{r2} \\ \vdots \\ \mathbf{x}_{ri} \\ \vdots \end{Bmatrix}, \mathbf{f}_f = \begin{Bmatrix} \mathbf{f}_{f1} \\ \mathbf{f}_{f2} \\ \vdots \\ \mathbf{f}_{fi} \\ \vdots \end{Bmatrix}, \mathbf{f}_r = \begin{Bmatrix} \mathbf{f}_{r1} \\ \mathbf{f}_{r2} \\ \vdots \\ \mathbf{f}_{ri} \\ \vdots \end{Bmatrix} \quad (2.5)$$

The matrices $\mathbf{M}_{rri}$ and $\mathbf{C}_{rri}$ are positive definite.

Bearings are modeled as spring-damper models connecting the individual body reference frames to ground. Assembling the bearings at this stage, they contribute the terms $\mathbf{C}_{rrB}$ to the damping matrix and $\mathbf{K}_{rrB}$ to the stiffness matrix:

$$\begin{bmatrix} \mathbf{M}_{ff} & \mathbf{M}_{fr} \\ \mathbf{M}_{rf} & \mathbf{M}_{rr} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}}_f \\ \ddot{\mathbf{x}}_r \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_{ff} & \mathbf{C}_{fr} \\ \mathbf{C}_{rf} & \mathbf{C}_{rr} + \mathbf{C}_{rrB} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{x}}_f \\ \dot{\mathbf{x}}_r \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{ff} & 0 \\ 0 & \mathbf{K}_{rrB} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_f \\ \mathbf{x}_r \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_f \\ \mathbf{f}_r \end{Bmatrix} \quad (2.6)$$

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f} \quad (2.7)$$

where:

$$\mathbf{x} = \begin{Bmatrix} \mathbf{x}_f \\ \mathbf{x}_r \end{Bmatrix}.\mathbf{f} = \begin{Bmatrix} \mathbf{f}_f \\ \mathbf{f}_r \end{Bmatrix}, \mathbf{M} = \begin{bmatrix} \mathbf{M}_{ff} & \mathbf{M}_{fr} \\ \mathbf{M}_{rf} & \mathbf{M}_{rr} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{C}_{ff} & \mathbf{C}_{fr} \\ \mathbf{C}_{rf} & \mathbf{C}_{rr} + \mathbf{C}_{rrB} \end{bmatrix}, \mathbf{K} = \begin{bmatrix} \mathbf{K}_{ff} & 0 \\ 0 & \mathbf{K}_{rrB} \end{bmatrix} \quad (2.8)$$

The term $\mathbf{K}_{rrB}$ is usually not of full rank. The stiffness matrix $\mathbf{K}$ is therefore not invertible.

In static and quasi-static analyses, the mass matrix $\mathbf{M}$ and the damping matrix $\mathbf{C}$ are neglected and the system 2.7 reduces to:

$$\mathbf{K}\mathbf{x} = \mathbf{f} \quad (2.9)$$

In a dynamic analysis, a time-discretization based on the Newmark method is used. Equation 2.7 when discretized becomes:

$$\begin{aligned} &\left(\mathbf{M} + \gamma\Delta t\mathbf{C} + \beta\Delta t^2\mathbf{K}\right)\mathbf{x}_{n+1} \\ &+ \left(-2\mathbf{M} + (1-2\gamma)\,\Delta t\mathbf{C} + \left(\frac{1}{2} - 2\beta + \gamma\right)\Delta t^2\mathbf{K}\right)\mathbf{x}_n \\ &+ \left(\mathbf{M} - (1-\gamma)\,\Delta t\mathbf{C} + \left(\frac{1}{2} + \beta - \gamma\right)\Delta t^2\mathbf{K}\right)x_{n-1} \\ &= \left(\beta.\mathbf{f}_{n+1} + \left(\frac{1}{2} - 2\beta + \gamma\right)\mathbf{f}_n + \left(\frac{1}{2} + \beta - \gamma\right)\mathbf{f}_{n-1}\right)\Delta t^2 \end{aligned} \quad (2.10)$$

where $\mathbf{x}_n = \mathbf{x}\,(t_o + n\Delta t)$ and $\mathbf{f}_n = \mathbf{f}\,(t_o + n\Delta t)$.

It can be shown that all schemes for which $\gamma = \frac{1}{2}$ and $\beta \geq \frac{1}{4}$ are unconditionally stable and show no artificial damping. Several commonly used integration schemes happen to be special cases of this 'three point' scheme for certain combinations of γ and β. They are tabulated in Table 2.1.

Rearranging terms,

Table 2.1: Common Three Point Integration Techniques

Method Name	β	γ
Central Difference	0	$\frac{1}{2}$
Backward Difference	1	$\frac{3}{2}$
Linear Acceleration	$\frac{1}{10}$	$\frac{1}{2}$
Galerkin	$\frac{4}{5}$	$\frac{3}{2}$
Fox Goodwin	$\frac{1}{12}$	$\frac{1}{2}$
Average Acceleration	$\frac{1}{4}$	$\frac{1}{2}$
Unconditionally stable with no artificial damping	$\geq \frac{1}{4}$	$\frac{1}{2}$

$$\begin{aligned}\left(\mathbf{M}+\gamma\Delta t\mathbf{C}+\beta\Delta t^2\mathbf{K}\right)\mathbf{x}_{n+1} = \\ &\left(\beta.\mathbf{f}_{n+1}+\left(\frac{1}{2}-2\beta+\gamma\right)\mathbf{f}_n+\left(\frac{1}{2}+\beta-\gamma\right)\mathbf{f}_{n-1}\right)\Delta t^2 \\ &-\left(-2\mathbf{M}+\left(1-2\gamma\right)\Delta t\mathbf{C}+\left(\frac{1}{2}-2\beta+\gamma\right)\Delta t^2\mathbf{K}\right)\mathbf{x}_n \\ &-\left(\mathbf{M}-\left(1-\gamma\right)\Delta t\mathbf{C}+\left(\frac{1}{2}+\beta-\gamma\right)\Delta t^2\mathbf{K}\right)x_{n-1}\end{aligned} \tag{2.11}$$

or

$$\widehat{\mathbf{K}}\hat{\mathbf{x}}=\hat{\mathbf{f}} \tag{2.12}$$

where

$$\widehat{\mathbf{K}}=\mathbf{M}+\gamma\Delta t\mathbf{C}+\beta\Delta t^2\mathbf{K} \tag{2.13}$$

is treated as an effective stiffness matrix,

$$\hat{\mathbf{x}}=\mathbf{x}_{n+1} \tag{2.14}$$

and

$$\begin{aligned}\hat{\mathbf{f}} = &\left(\beta.\mathbf{f}_{n+1}+\left(\frac{1}{2}-2\beta+\gamma\right)\mathbf{f}_n+\left(\frac{1}{2}+\beta-\gamma\right)\mathbf{f}_{n-1}\right)\Delta t^2 \\ &-\left(-2\mathbf{M}+\left(1-2\gamma\right)\Delta t\mathbf{C}+\left(\frac{1}{2}-2\beta+\gamma\right)\Delta t^2\mathbf{K}\right)\mathbf{x}_n \\ &-\left(\mathbf{M}-\left(1-\gamma\right)\Delta t\mathbf{C}+\left(\frac{1}{2}+\beta-\gamma\right)\Delta t^2\mathbf{K}\right)x_{n-1}\end{aligned} \tag{2.15}$$

is an effective load vector.

Equation 2.12 is identical in form to the static equation 2.9, and in the discussion below, they are treated identically.

When the matrix $\mathbf{K}$ is singular, a coordinate transformation $\mathbf{x}\rightarrow\left\{\begin{array}{c}\mathbf{x}_\phi\\ \mathbf{x}_\theta\end{array}\right\}$ is used to separate out the non-singular part of the system equation. The coordinate transformation chosen is linear:

$$\mathbf{x}=\left[\begin{array}{cc}\mathbf{T}_\phi & \mathbf{T}_\theta\end{array}\right]\left\{\begin{array}{c}\mathbf{x}_\phi\\ \mathbf{x}_\theta\end{array}\right\}=\mathbf{T}\left\{\begin{array}{c}\mathbf{x}_\phi\\ \mathbf{x}_\theta\end{array}\right\} \tag{2.16}$$

$$\mathbf{T}^T\mathbf{f}=\left[\begin{array}{c}\mathbf{T}_\phi^T\\ \mathbf{T}_\theta^T\end{array}\right]\mathbf{f}=\left\{\begin{array}{c}\mathbf{f}_\phi\\ \mathbf{f}_\theta\end{array}\right\} \tag{2.17}$$

The system equation 2.9 or 2.12 is transformed into:

$$\mathbf{T}^T\mathbf{K}\mathbf{T}\left\{\begin{array}{c}\mathbf{x}_\phi\\ \mathbf{x}_\theta\end{array}\right\}=\left\{\begin{array}{c}\mathbf{f}_\phi\\ \mathbf{f}_\theta\end{array}\right\} \tag{2.18}$$

or,

$$\left[\begin{array}{cc}\mathbf{K}_{\phi\phi} & 0\\ 0 & 0\end{array}\right]\left\{\begin{array}{c}\mathbf{x}_\phi\\ \mathbf{x}_\theta\end{array}\right\}=\left\{\begin{array}{c}\mathbf{f}_\phi\\ \mathbf{f}_\theta\end{array}\right\} \tag{2.19}$$

The transformation is similar to one that would be used to diagonalize the system equation. The matrix $\mathbf{K}_{\phi\phi}$ is diagonal. The partitioning of $\mathbf{x}$ into $\mathbf{x}_\phi$ and $\mathbf{x}_\theta$ is set up so that all the diagonal terms in $\mathbf{K}_{\phi\phi}$ are non zero. $\mathbf{K}_{\phi\phi}$ is therefore positive definite. For a dynamic situation, the effective stiffness matrix $\widehat{\mathbf{K}}$ is always positive definite, so that $\mathbf{x}_\theta$ would have a dimension of zero.

The upper partition of 2.19 gives:

$$\mathbf{K}_{\phi\phi}\mathbf{x}_\phi=\mathbf{f}_\phi \tag{2.20}$$

or

$$\mathbf{x}_\phi=\mathbf{K}_{\phi\phi}^{-1}\mathbf{f}_\phi=\mathbf{K}_{\phi\phi}^{-1}\mathbf{T}_\phi\mathbf{f} \tag{2.21}$$

and the lower partition yields:

$$\mathbf{f}_\theta=0 \tag{2.22}$$

Using 2.17,

$$\mathbf{T}_\theta^T\mathbf{f}=0 \tag{2.23}$$

Candidate Point Pairs (CPPs) with surface normals along a common axis are calculated by a search algorithm with specified 'separation' tolerance. The tooth surface is defined with arbitrary precision as either a continuous curve or a set of surface coordinates with specified surface normal (these surface coordinates are not limited to points on the finite element mesh). The (arbitrary) number of points used to describe the surface dictates the pool of possible contact points. Define the following vector quantities: $\hat{\varepsilon}$ = separation distances of all CPPs along their common normal at some instant in the unloaded and undeformed state, $\mathbf{d}$ = separations of the CPPs along their common normal in the deformed state, δ = changes in separation due to the load, and $\mathbf{p}$ = compressive loads acting along the normals of each CPP. The final separation of the CPPs is given by

$$\mathbf{d}=\hat{\varepsilon}+\delta \tag{2.24}$$

The external load vector $\mathbf{f}$ in 2.23 is related to the vector $\mathbf{p}$ of contact forces by a linear relationship:

$$\mathbf{f}=\mathbf{E}\mathbf{p}+\mathbf{f}_o \tag{2.25}$$

Here $\mathbf{E}$ is a known matrix, and $\mathbf{f}_o$ is a vector of known non-contact external loads on the model. The increase in separation δ is related to displacement vector:

$$\delta=\left[\begin{array}{cc}\mathbf{F}_\phi & \mathbf{F}_\theta\end{array}\right]\left\{\begin{array}{c}\mathbf{x}_\phi\\ \mathbf{x}_\theta\end{array}\right\} \tag{2.26}$$

$\mathbf{F}_\phi$ and $\mathbf{F}_\theta$ are known matrices. From 2.21 and 2.25,

$$\mathbf{x}_\phi=\mathbf{K}_{\phi\phi}^{-1}\mathbf{T}_\phi^T\left(\mathbf{E}\mathbf{p}+\mathbf{f}_o\right) \tag{2.27}$$

or

$$\mathbf{x}_\phi=\left(\mathbf{K}_{\phi\phi}^{-1}\mathbf{T}_\phi^T\mathbf{E}\right)\mathbf{p}+\left(\mathbf{K}_{\phi\phi}^{-1}\mathbf{T}_\phi^T\right)\mathbf{f}_o \tag{2.28}$$

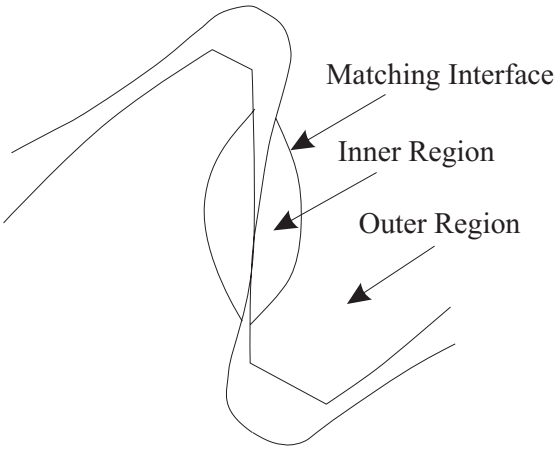

Figure 2.1: The matching interface for the near-field and far-field regions.

using 2.26,

$$\delta = \mathbf{F}_\phi \mathbf{x}_\phi + \mathbf{F}_\theta \mathbf{x}_\theta \tag{2.29}$$

Equation 2.29 is a compliance relationship that contains only the finite element contribution (outer region of Figure 2.1) to the increase in separation at the contact points. In addition to this, a contribution from a local deformation field is superimposed. This contribution is calculated using a semi-analytical solution near the contact zone (inner region of Figure 2.1) [6, 4]. This introduces an additional term $\mathbf{A}_{local}\mathbf{p}$ into the compliance relationship 2.29 and we get:

$$\delta = \mathbf{F}_\phi \mathbf{x}_\phi + \mathbf{F}_\theta \mathbf{x}_\theta + \mathbf{A}_{local}\mathbf{p} \tag{2.30}$$

$\mathbf{A}_{local}$ is the local deformation component of compliance. Therefore:

$$\delta = \left(\mathbf{F}_\phi \mathbf{K}_{\phi\phi}^{-1} \mathbf{T}_\phi^T \mathbf{E} + \mathbf{A}_{local}\right) \mathbf{p} + \left(\mathbf{F}_\phi \mathbf{K}_{\phi\phi}^{-1} \mathbf{T}_\phi^T \mathbf{f}_o\right) + \mathbf{F}_\theta \mathbf{x}_\theta \tag{2.31}$$

Substituting in 2.24, we get:

$$\mathbf{d} = \left(\mathbf{F}_\phi \mathbf{K}_{\phi\phi}^{-1} \mathbf{T}_\phi^T \mathbf{E} + \mathbf{A}_{local}\right) \mathbf{p} + \mathbf{F}_\theta \mathbf{x}_\theta + \left(\hat{\varepsilon} + \mathbf{F}_\phi \mathbf{K}_{\phi\phi}^{-1} \mathbf{T}_\phi^T \mathbf{f}_o\right) \tag{2.32}$$

or,

$$\mathbf{d} = \mathbf{A}\mathbf{p} + \mathbf{C}\mathbf{x}_\theta + \varepsilon \tag{2.33}$$

where

$$\mathbf{C} = \mathbf{F}_\theta \tag{2.34}$$

$$\mathbf{A} = \mathbf{F}_\phi \mathbf{K}_{\phi\phi}^{-1} \mathbf{T}_\phi^T \mathbf{E} + \mathbf{A}_{local} \tag{2.35}$$

and

$$\varepsilon = \hat{\varepsilon} + \mathbf{F}_\phi \mathbf{K}_{\phi\phi}^{-1} \mathbf{T}_\phi^T \mathbf{f}_o \tag{2.36}$$

In addition, 2.23 and 2.25 lead to the following 'equilibrium equation' which must be satisfied

$$\mathbf{T}_\theta \left(\mathbf{E}\mathbf{p} + \mathbf{f}_o\right) = 0 \tag{2.37}$$

or

$$\mathbf{T}_\theta \mathbf{E}\mathbf{p} + \mathbf{T}_\theta \mathbf{f}_o = 0 \tag{2.38}$$

which is of the form:

$$\mathbf{Bp} = \lambda \tag{2.39}$$

where $\mathbf{B} = \mathbf{T}_\theta \mathbf{E}$ and $\lambda = -\mathbf{T}_\theta \mathbf{f}_o$

The equations 2.33 and 2.39, together with compatibility conditions are posed as a 'contact problem':

$$\begin{array}{ll} \text{Solve} & \left\{ \begin{array}{l} \mathbf{d} = \mathbf{Ap} + \mathbf{Cx}_\theta + \varepsilon \\ \mathbf{Bp} = \lambda \end{array} \right. \\ \text{for} & \mathbf{d},\ \mathbf{p} \text{ and } \mathbf{x}_\theta \\ \text{subject to} & \mathbf{d} \geq \mathbf{0},\ \mathbf{p} \geq \mathbf{0} \text{ and } \mathbf{d}^T\mathbf{p} = 0 \end{array} \tag{2.40}$$

This problem can be solved as described in Section 2.2. The calculated contact loads $\mathbf{p}$ are then used in the combined surface integral/finite element solution to calculate tooth deflections and stresses. This process is repeated at each integration time step. The matrices $\mathbf{B}$ and $\mathbf{C}$, which depend on straightforward kinematics, are recalculated at each step as the bodies undergo the specified rigid body motions. While $\mathbf{A}$ is also recalculated at each step because of the changing contact conditions, the finite element stiffness matrix that combines with the semi-analytical tooth surface model in assembling $\mathbf{A}$ is determined only once and does not require updating.

The combined surface integral/finite element solution is described in [4] with a simplified discussion in [6]. In essence, the concept is to match an analytical 'inner' solution that applies near the tooth surface (calculated from the solution for a point load on a half-space) with an 'outer' finite element solution that applies slightly away from the tooth surface where the displacement gradients are less steep. Because the solution in the 'inner' region at the tooth surface does not depend on finite elements to calculate the tooth surface deformations, model the contact mechanics, or define the geometry via node points, the need for an extremely refined mesh is removed. This key point makes dynamic analysis with careful contact modeling possible for sufficient number of time steps to obtain frequency domain response calculations. Note that conventional, linear, four-node finite elements are used away from the tooth surface.

A crucial distinction of this formulation is that no a priori assumptions about the nature of the dynamic excitation are specified. The time-varying mesh stiffness and/or static transmission error excitation that are required in virtually all existing models (e.g., [9, 13]) are calculated quantities, not specified inputs that are necessarily approximate. Here, only the operating torque and speed are inputs for the dynamic analysis. The desired outputs are the rotational vibrations of each body, and the net tooth contact force calculated from the contact force vector at each meshing tooth.

2.2 The Contact Algorithm

Previously, we used a technique based on the simplex algorithm [3]. We have now implemented a solver based on the Quadratic Programming Technique, which is more efficient[8, 9]. Its convergence is also more robust.

To solve the contact problem 2.40, consider the Quadratic programming problem:

$$\begin{array}{ll} \text{Maximize} & \mathbf{x}^T\mathbf{Dx} - \mathbf{c}^T\mathbf{x} \\ \text{w.r.t.} & \mathbf{x} \\ \text{subject to} & \mathbf{x} \geq \mathbf{0} \text{ and } \mathbf{Ax} = \mathbf{b} \end{array} \tag{2.41}$$

Here $\mathbf{A}$ is an $m \times n$ matrix, and $\mathbf{D}$ is a symmetric negative definite $n \times n$ matrix. It can be shown using the Karush Kuhn-Tucker condition for optimality that this quadratic programming

problem is equivalent to the following problem:

$$\begin{array}{ll} \text{Solve} & \left\{ \begin{array}{rcl} \mathbf{Ax} & = & \mathbf{b} \\ 2\mathbf{D} - \mathbf{C}\lambda + \mathbf{v} & = & \mathbf{c} \end{array} \right. \\ \text{for} & \mathbf{v},\ \mathbf{x} \text{ and } \lambda \\ \text{subject to} & \mathbf{x} \geq \mathbf{0},\ \mathbf{v} \geq 0, \text{ and } \mathbf{x}^T\mathbf{v} = 0 \end{array} \tag{2.42}$$

Comparing 2.42 with 2.40, we see that the two are identical, apart from a renaming of variables and matrices.

The first step in solving the system is to obtain a Basis Feasible Solution $\mathbf{x}_b$ to $\mathbf{Ax} = \mathbf{b}$. Several techniques are available. We find the most convenient method is to use only Phase I of a revised simplex solver to obtain this basic feasible solution. This is a relatively quick step, because m is much smaller than n. If the revised simplex solver fails to find the necessary basic feasible solution, then it means that the original contact problem had some rigid body type degrees of freedom that could not be equilibrated by any of the available contact constraints.

Let $\mathbf{B}$ be the sub-matrix of $\mathbf{A}$ that corresponds to the columns that are in this basis. The inverse $\mathbf{B}^{-1}$ is also readily obtained from the tableau of the revised simplex solver at the time of its termination.

We now set up the following linear program:

$$\begin{array}{ll} \text{Minimize} & (u_1 + u_2 + u_3 + \ldots + u_n) \\ \text{w.r.t.} & \mathbf{x},\ \lambda^+,\ \lambda^-,\ \mathbf{v} \text{ and } \mathbf{u} \\ \text{subject to} & \left[\begin{array}{ccccc} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2\mathbf{D} & -\mathbf{C} & +\mathbf{C} & \mathbf{I}_n & \mathbf{E} \end{array} \right] \left\{ \begin{array}{c} \mathbf{x} \\ \lambda^+ \\ \lambda^- \\ \mathbf{v} \\ \mathbf{u} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{b} \\ \mathbf{c} \end{array} \right\} \\ & \mathbf{x} \geq \mathbf{0},\ \lambda^+ \geq \mathbf{0},\ \lambda^- \geq \mathbf{0},\ \mathbf{v} \geq \mathbf{0} \text{ and } \mathbf{u} \geq \mathbf{0} \end{array} \tag{2.43}$$

Here

$$\mathbf{E} = \text{Diag}(\mathbf{c} - 2\mathbf{D}_b\mathbf{x}_b) \tag{2.44}$$

If we set up a revised simplex solver to solve 2.43, then an initial basic feasible solution is set up as:

$$\left\{ \begin{array}{c} \mathbf{x}_b \\ \mathbf{u} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{x}_b \\ \left\{ \begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \end{array} \right\} \end{array} \right\} \tag{2.45}$$

The corresponding initial tableau for the revised simple solver is:

$$\mathbf{T} = \left[\begin{array}{ccc} \mathbf{B}^{-1} & \mathbf{0} \quad \mathbf{0} & \mathbf{x}_b \\ -\mathbf{E}2\mathbf{D}\mathbf{B}^{-1} & \mathbf{E} \quad \mathbf{0} & \left\{ \begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \end{array} \right\} \\ * & * \quad 1 & * \end{array} \right] \tag{2.46}$$

The terms in the last row of 2.46 that are marked by an asterisk are initialized to the sum of the n rows immediately above them.

With this initial tableau, the standard revised simplex algorithm is used to minimize the cost variable. At every pivoting step, one restriction is placed while selecting the pivot column

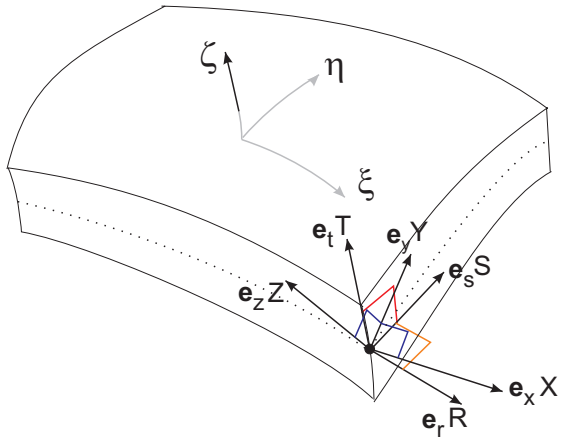

Figure 2.2: The orthogonal unit vectors associated with the two coordinate systems at a modal point of a shell finite element.

entering the active basis: If x_i is already in the basis, then v_i is only allowed into the basis if it replaces x_i. Similarly, if v_i is in the basis, then x_i is only allowed to enter the basis if it is going to replace v_i.

We have found that the revised simplex algorithm converges after approximately $3n$ pivoting steps.

Convergence is robust as long as $\mathbf{D}$ is well conditioned, and a solution to the problem exists. We have found that scaling the original equations so that all diagonal elements of $\mathbf{D}$ are of the same magnitude helps in the improving the robustness of the algorithm.

2.3 Shell Elements

Calyx has been developed developed for the analysis of a variety of geared transmission systems. In aerospace and other lightweight transmission applications, the housing tends to be flexible, and an accurate analysis of the gear contact requires that a good housing model be used. Modeling these housings efficiently requires shell elements. Incorporating shell elements into *Calyx* was one of our SBIR project objectives.

2.3.1 Implementation

Correctly implementing a shell finite element requires care in the selection of the material deformation equations, the displacement interpolation functions, and the integration technique.

Figure 2.2 shows an isoparametric shell element. The coordinates are interpolated in a manner identical to that of regular solid elements. Figure 2.2 shows a displacement node located on the mid-surface of the element. Also shown are two sets of orthogonal unit vectors. The first set $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$ is oriented along the global X, Y and Z axes. The second set consists of $\mathbf{e}_r$, $\mathbf{e}_s$ and $\mathbf{e}_t$. Of these, $\mathbf{e}_r$ and $\mathbf{e}_s$ are tangent to the mid-surface and perpendicular to each other. The third unit vector $\mathbf{e}_t$ is normal to the mid-surface. The conventional shell element formulation (Zienkiewicz[10] and Bathe[11]) uses a displacement interpolation that has at each node, three translational degrees of freedom u_x, u_y and u_z, and two rotations θ_r and θ_s about the axes R and S , respectively, as shown in Figure 2.3.

We prefer not to use the rotational degrees of freedom. Instead, we use the derivatives of the displacement field with respect to the local coordinate ζ as shown in Figure 2.4. The displacement

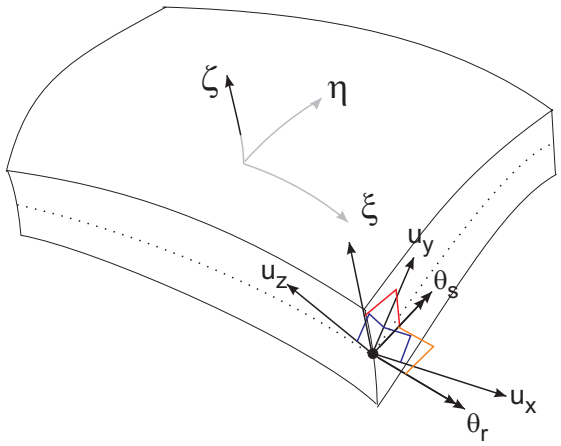

Figure 2.3: The three translational and two rotational degrees of freedom normally associated with a shell finite element.

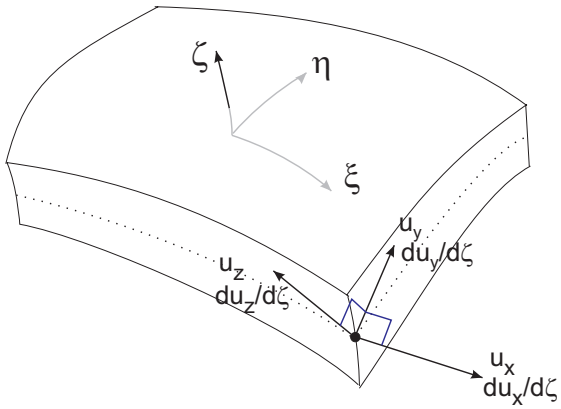

Figure 2.4: The three translational degrees of freedom and the degrees of freedom associated with the derivatives of the displacement field with respect to the thickness local coordinate z.

field obtained by using these degrees of freedom spans the entire field of Zienkiewicz[10] and Bathe[11], but in addition, is able to represent normal strain in the thickness direction. We constrain this normal strain to zero by applying appropriate nodal constraints. With these constraints, our displacement field is identical to that of Zienkiewicz and Bathe. However, by not using rotations as primary nodal degrees of freedom, the implementation of the finite element is considerably simplified. We use reduced integration to improve the performance of the finite elements as recommended by Zienkiewicz[10] and Bathe[11].

2.3.2 Testing the Shell Element on a Flat Plate Problem

A flat plate example is ideal for testing the bending response of the shell element. In the flat plate, the curvature is zero, and so the membrane forces are completely decoupled from the bending stresses. An 'Exact' solution for a rectangular flat plate is available in Timoshenko and Woinkowsky-Krieger [12]. A square plate is chosen for this study. The basic data for this plate is shown in Table 2.2.

Table 2.2: Flat Plate Data

Length of plate (a)	10.0 inch
Width of plate (b)	10.0 inch
Thickness (h)	0.1 inch
Young's Modulus (E)	30.0e6 psi
Poisson's Ratio ν	0.3
Pressure applied on upper surface (q)	100 psi

The deflection at center of the plate for a square plate ($a = b$), for the clamped case is given by[12]:

$$w = 0.00126\frac{qa^4}{D} = 0.45864inch \tag{2.47}$$

For the simply supported case, the deflection at center of the square is

$$w = 0.00406\frac{qa^4}{D} = 1.47784inch \tag{2.48}$$

Here D is the flexural modulus given by:

$$D = \frac{Eh^3}{12\left(1-v^2\right)} = 2747.25lbf - inch \tag{2.49}$$

Figure 2.5 shows a finite element model of this square plate. The model has 16 elements along the length and width of the plate.

Table 2.3 shows the center deflection computed by the finite element model for various mesh densities and element types using a full order Gauss quadrature. As expected, the linear finite element shows poor convergence for the clamped boundary condition, and the finite element system is singular for the simply supported boundary condition. The quadratic element shows good convergence for both the boundary conditions. The cubic element shows the best convergence behavior.

Table 2.4 shows the results using reduced integration. In reduced integration, the order of gauss quadrature is one less than what is required for the finite element. Both Zienkiewicz[10] and Bathe[11] recommend using reduced integration as a means of speeding up convergence. We find that using the linear shell element always led to singular stiffness matrices with reduced

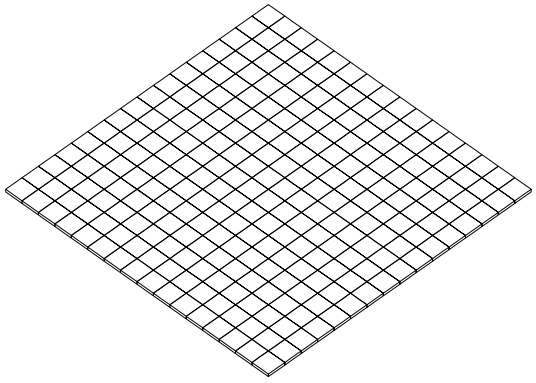

Figure 2.5: Finite element model of a uniformly loaded square flat plate.

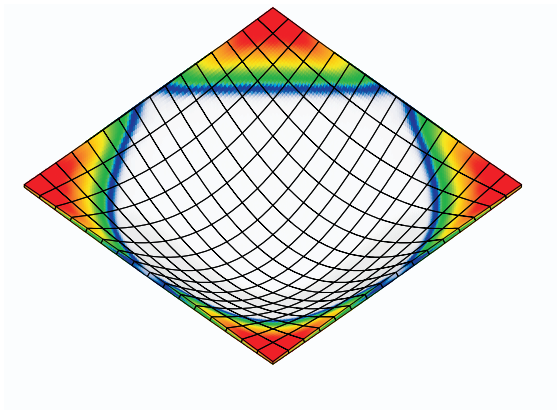

Figure 2.6: Deformed finite element model of a uniformly loaded square flat plate simply supported along the boundary. The deformation is exaggerated by a factor of 2.

integration. For the simply supported case, the stiffness matrices for all element types were either singular or almost singular. For quadratic and cubic elements, when they were not singular, the results predicted were very good. For the clamped boundary condition, stiffness matrix singularity was not a problem. The convergence observed in this case was much better than observed with full integration.

We see that with full integration the performance of the linear element is poor at best, and with reduced integration the linear element is always singular. Better performance can be obtained by using selectively reduced integration. Here a full integration order is used for the bending terms of the stiffness matrix and a reduced order is used for the shear terms. The results obtained using selectively reduced integration are shown in Table 2.5.

Figure 2.6 shows a deformed geometry plot of the simply supported plate. Maximum stresses occur near the corners of the plate. Figure 2.7 shows the deformed shape and stresses in the clamped plate. Here the stresses at the corners are very small. The bending stresses are highest at the middle of the four edges.

We conclude that the shell element implementation has successfully passed the bending test. Also, we have decided not to use reduced integration in *Calyx*, because of its tendency to produce singular stiffness matrices. We use selectively reduced integration instead.

Table 2.3: Plate deflection using full integration (Exact solution: simply-supported =1.47784, clamped =0.45864)

Shell Element Type	No. of Gauss Points	No. of elements	Deflection for simply supported boundary condition	Deflection for clamped boundary condition
Linear	2x2x2	2x2	Singular	Singular
		4x4	Singular	Singular
		8x8	Singular	0.013364
		16x16	Singular	0.0481433
Quadratic	3x3x2	2x2	1.29593	0.0055
		4x4	1.45161	0.36333
		8x8	1.475848	0.4385
		16x16	1.4833	0.45705
Cubic	4x4x2	2x2	1.50231	0.482619
		4x4	1.48291	0.461334
		8x8	1.484188	0.46146
		16x16	1.48757	0.4614497

2.3.3 Testing the Shell Element on a Hemispherical Shell Problem

Unlike the flat plate example, a hemispherical shell has membrane forces and bending moments that are coupled. Closed form solutions are available when a hemispherical shell is subjected to uniform pressure and is clamped at the edge. It is therefore a good example for the validation of the coupling effects between the membrane forces and bending moments.

Under uniform pressure, a spherical shell simply is compressed to a smaller radius, without creating any bending. However, if the sphere is not complete, and is constrained from radial motion at the edge (Figure 2.8), then bending moments are created near the edge. These bending moments die away as we move away from the edge. The thinner the shell, the faster these bending moments die off.

Table 2.6 shows the data for the hemispherical shell example. An analytical solution for the problem of a spherical shell under uniform load can be obtained by the linear superposition of a particular solution and a homogenous solution, as described by Timoshenko and Woinkowsky-Krieger[12].

Figure 2.9 shows the finite element model used here. The model represents a quadrant of the hemisphere. Along the two meridian edges, the nodes are allowed to move in a radial and axial direction, but are constrained against translation along the latitude of the sphere. At the clamped edge, the nodes are completely constrained. 192 cubic elements are used in this mesh.

Figure 2.10 shows the deformed finite element model. It is clear that away from the boundary, the shape of the shell remains spherical, and the bending stresses induced are very small. Near the boundary, significant bending stresses are generated.

Figure 2.11 shows a comparison of the bending moment in a meridional cross section, as predicted by the analytical model with the prediction of the finite element model. Agreement is excellent. Figure 2.12 shows a comparison of the membrane force acting in the meridional direction. Figures 2.13 and 2.14 show the comparison of results with selectively reduced integration. Again the agreement is excellent, with the maximum error being about 4%.

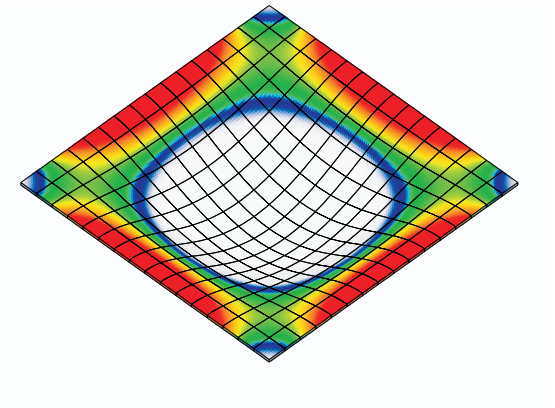

Figure 2.7: Deformed finite element model of a uniformly loaded square flat plate clamped along the boundary. The deformation is exaggerated by a factor of 4.

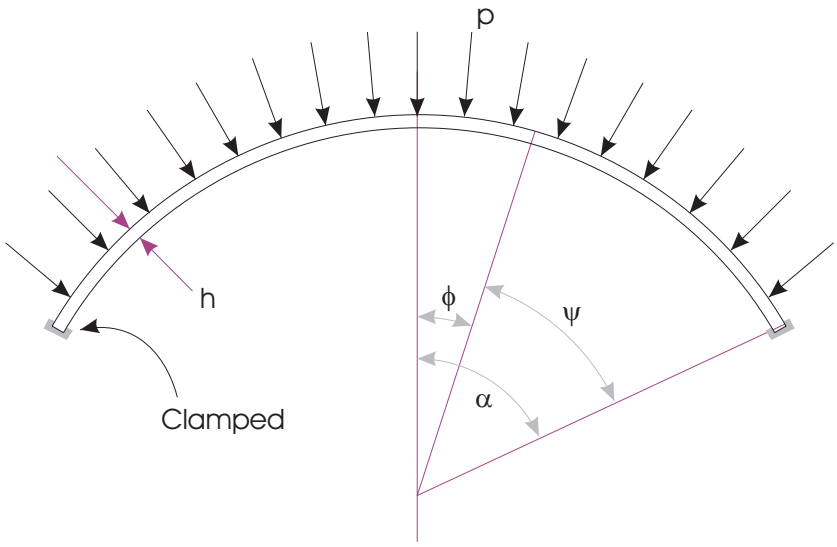

Figure 2.8: A hemispherical shell subjected to uniform pressure, and clamped at the edge.

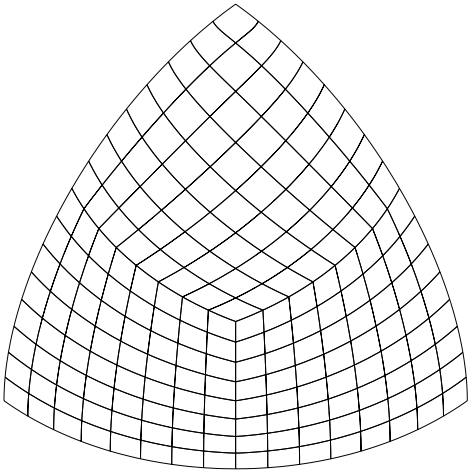

Figure 2.9: Finite element mesh used to model one quadrant of a hemispherical shell. The mesh has 192 cubic shell elements. Each shell element has 32 nodes. 16 modes are used for translation and 16 nodes for rotation.

Figure 2.10: Deformed finite element model of a uniformly loaded hemispherical shell clamped along the edge. The deformation is exaggerated by a factor of 2000.

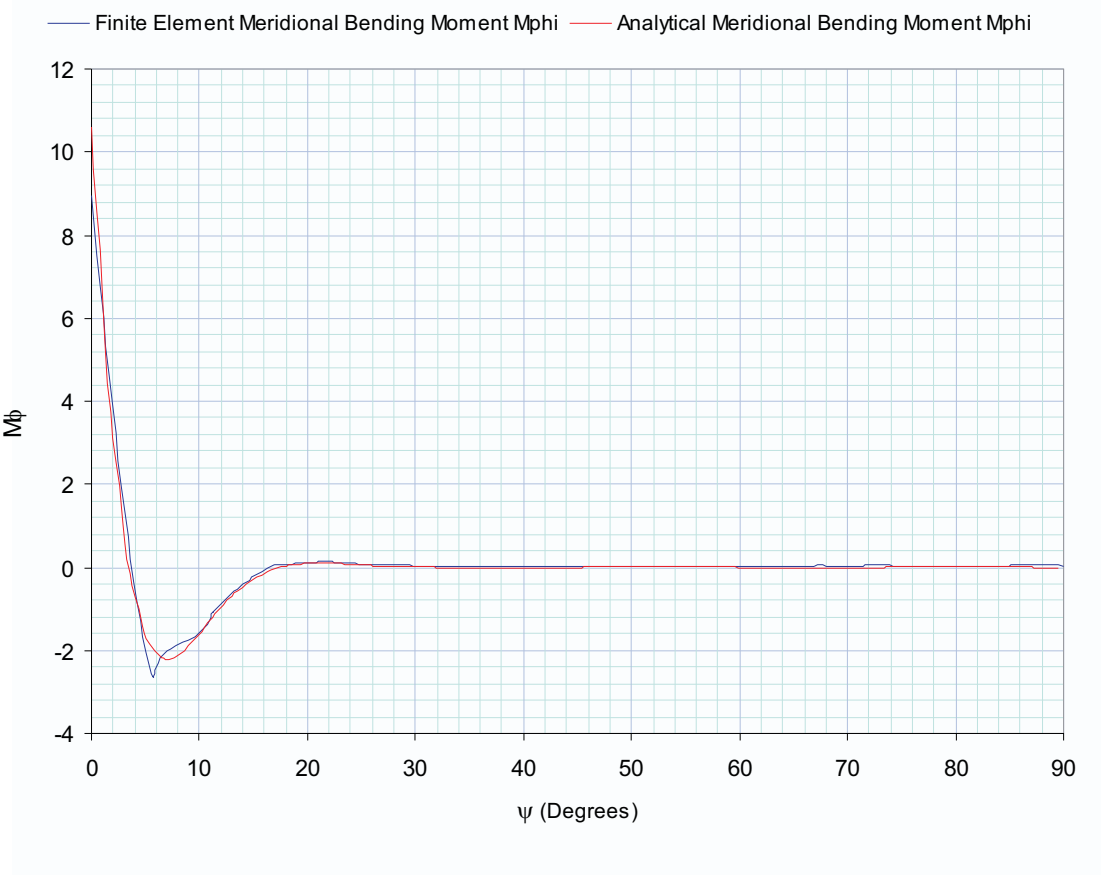

Figure 2.11: Comparison of meridional bending moment M_ϕ obtained by the finite element model with the analytical solution for a uniformly loaded hemispherical shell clamped along the edge.

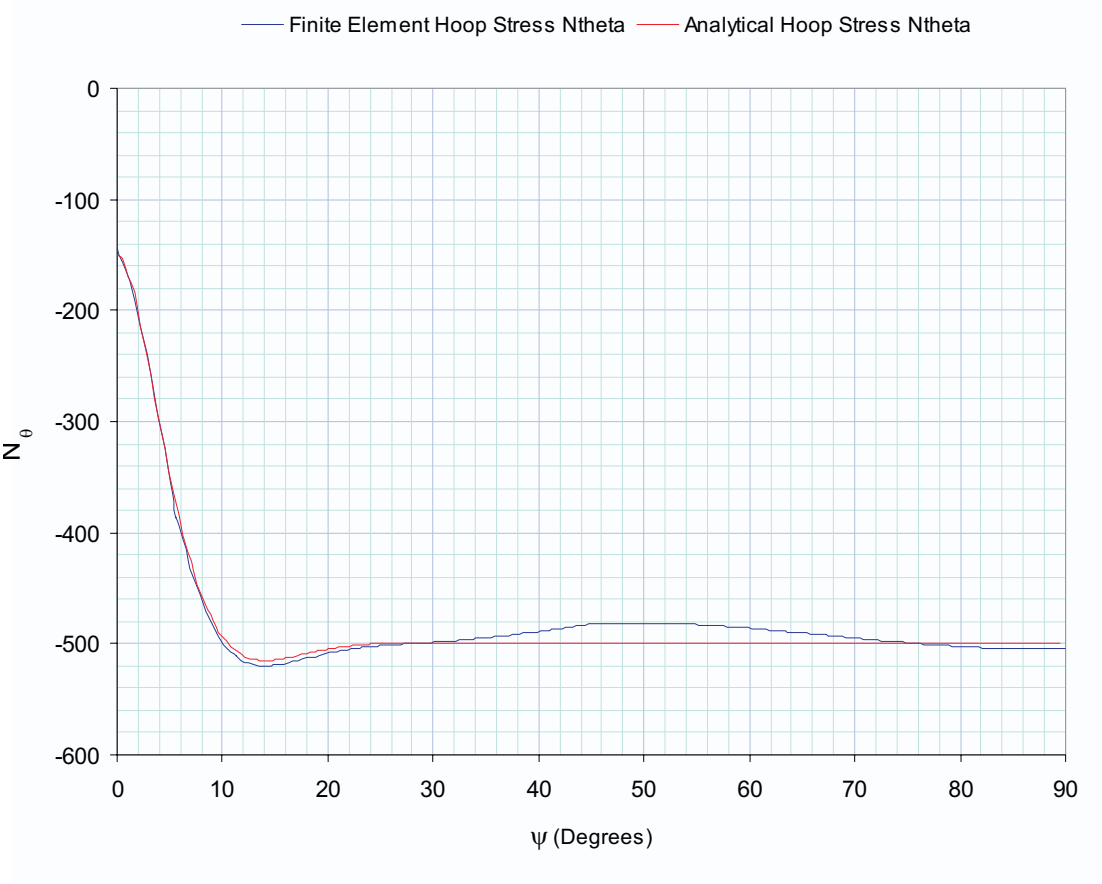

Figure 2.12: Comparison of hoop stress N_θ obtained by the finite element model with the analytical solution for a uniformly loaded hemispherical shell clamped along the edge.

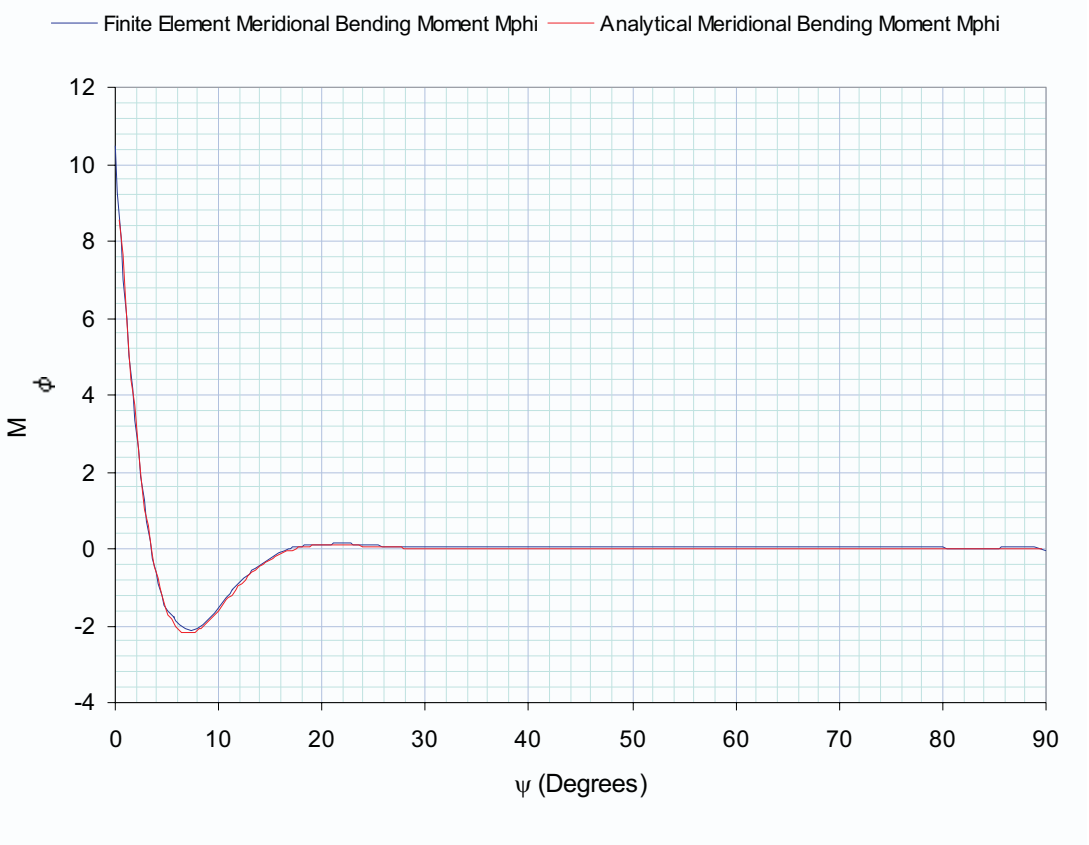

Figure 2.13: Comparison of meridional bending moment M_ϕ obtained by the finite element model (with selectively reduced integration order) with the analytical solution for a uniformly loaded hemispherical shell clamped along the edge.

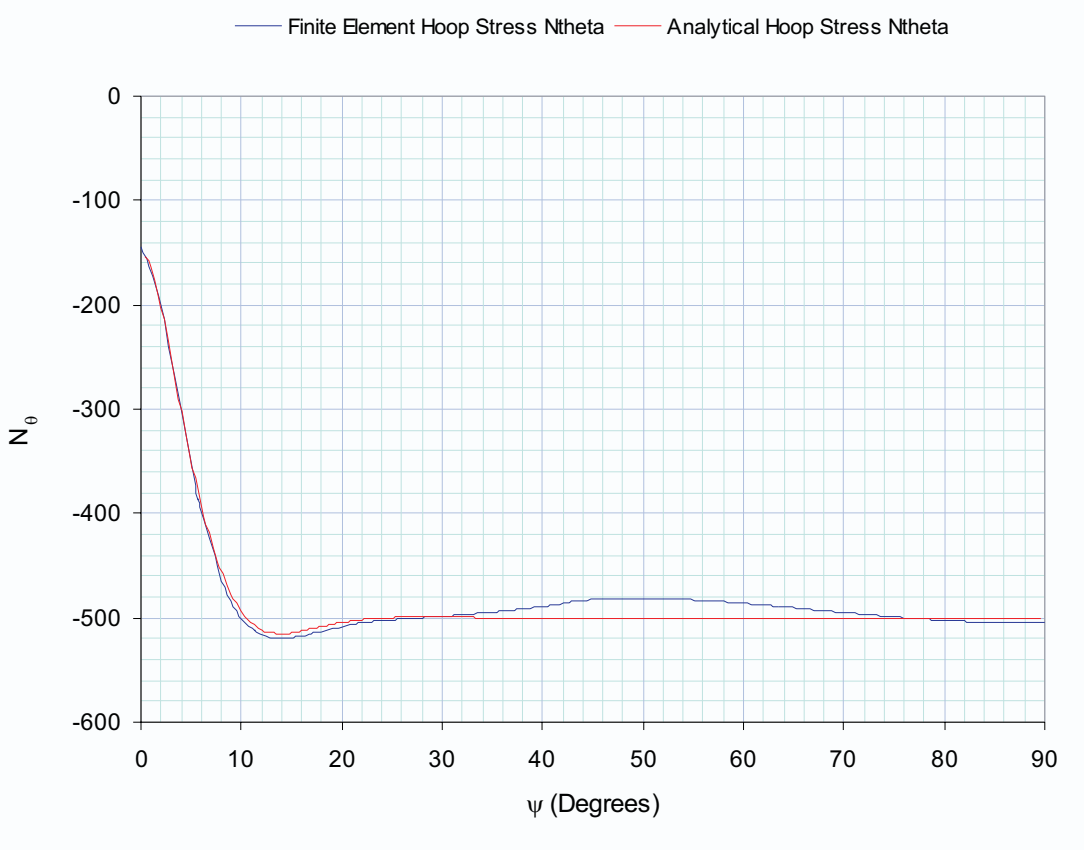

Figure 2.14: Comparison of hoop stress N_θ obtained by the finite element (with selectively reduced integration order) model with the analytical solution for a uniformly loaded hemispherical shell clamped along the edge.

Table 2.4: Plate deflection using reduced integration (Exact solution: simply-supported =1.47784, clamped =0.45864)

Shell Element Type	No. of Gauss Points	No. of elements	Deflection for simply supported boundary condition	Deflection for clamped boundary condition
Linear	1x1x2	2x2	Singular	Singular
		4x4	Singular	Singular
		8x8	Singular	Singular
		16x16	Singular	Singular
Quadratic	2x2x2	2x2	Singular	0.569533
		4x4	Singular	0.466517
		8x8	Singular	0.461802
		16x16	1.49216	0.461519
Cubic	3x3x2	2x2	Singular	0.4588
		4x4	Singular	0.46145
		8x8	1.49234	0.4615
		16x16	Singular	0.4615

2.4 Spiral Bevel and Hypoid Gear Modeling

Spiral bevel and hypoid gears are critical components of most helicopter transmissions and automobile rear axles. Modeling these is important to the success of our SBIR project.

Unlike spur, helical and straight bevel gears, there is no analytical description of the surface of a spiral-bevel or hypoid gear. The only complete definition of the gear surface is in terms of the kinematics of the machine used to cut them, the machine settings, and the cutter geometry. In order to build a *Calyx* model of a spiral-bevel or hypoid gear set, we need to model the kinematics of the cutting machine and the geometry of the cutter. Two techniques are commonly used in the industry for cutting spiral-bevel and hypoid gears.

The *Face Milling* method involves a milling type cutter. In this method, the cutter completes the generation of one side of a tooth, or the space between one pair of teeth. Then the tool is withdrawn from the workpiece, the workpiece is indexed by one tooth, the tool is fed back into this new position, and the process is repeated. There are two variations of the face-milling process. The 'Formate' face-milling process is used in automotive ring gears. In this process, there is no generating motion, and is faster. The 'Generated' face-milling process has generating motion, and is used for automotive pinions and for both the pinion and ring gear in aerospace applications.

In the *Face Hobbing* method the cutter rotation is synchronized with the rotation of the workpiece. While the workpiece rotates through an angle corresponding to one tooth, the cutter rotates by an angle corresponding to one cutter blade. The indexing motion is smooth, so the process is also referred to as *Continuous Indexing*. Two variations of the Face-Hobbing process are used. While cutting the ring gear, no generating motion is used. This reduces the cutting time. The mating pinion, on the other hand, is cut with generating motion.

The advantage that the face-milling process has is that it can be used for cutting as well as grinding. On the other hand, the face-hobbing process is more efficient because it needs fewer machine setups than face-milling. Its disadvantage is that it is not possible to grind a gear pair generated by this process. For this reason, face-milling is almost always used in aerospace applications where the gears need to be ground. In automotive applications, machining time is

Table 2.5: Plate deflection using selectively reduced integration (Exact solution: simply-supported =1.47784, clamped =0.45864)

Shell Element Type	No. of Gauss Points	No. of elements	Deflection for simply supported boundary condition	Deflection for clamped boundary condition
Linear	1x1x2	2x2	2.1888	0.0013
		4x4	1.575347	0.4418345
		8x8	1.505121	0.4562016
		16x16	1.491377	0.4601705
Quadratic	2x2x2	2x2	1.681203	0.5616886
		4x4	1.524404	0.4654262
		8x8	1.494692	0.4617201
		16x16	1.492135	0.461514
Cubic	3x3x2	2x2	1.555512	0.4577867
		4x4	1.502485	0.4614333
		8x8	1.492312	0.4614996
		16x16	1.492136	0.4615005

Table 2.6: Plate deflection using selectively reduced integration (Exact solution: simply-supported =1.47784, clamped =0.45864)

Radius of mid-surface of shell (a)	10.0 inch
Thickness (h)	0.1 inch
Angle a	90 Deg
Young's Modulus (E)	30.0e6 psi
Poisson's Ratio n	0.3
Pressure applied on upper surface (p)	100 psi

more of a concern than surface finish, and so face-hobbing is predominant.

We will provide brief descriptions of the face-milling and face-hobbing process that we have implemented.

2.4.1 Notation

In the discussion of machine kinematics, the symbols $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ will stand for the three unit vectors parallel to the X, Y and Z axes, respectively.

We will describe the kinematics of the spiral-bevel and hypoid gear cutting machines in terms of two transformation operators.

The operator $\mathcal{T}(\mathbf{v})$ takes one argument; a translation vector $\mathbf{v}$. The notation:

$$\mathbf{X}_2 = \mathcal{T}(\mathbf{v})\mathbf{X}_1 \tag{2.50}$$

means that the reference frame $\mathbf{X}_2$ is obtained by translating the origin of $\mathbf{X}_1$ by the vector $\mathbf{v}$. $\mathbf{v}$ may be measured either in $\mathbf{X}_1$ or $\mathbf{X}_2$. Coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$ of a vector measured in $\mathbf{X}_1$ and

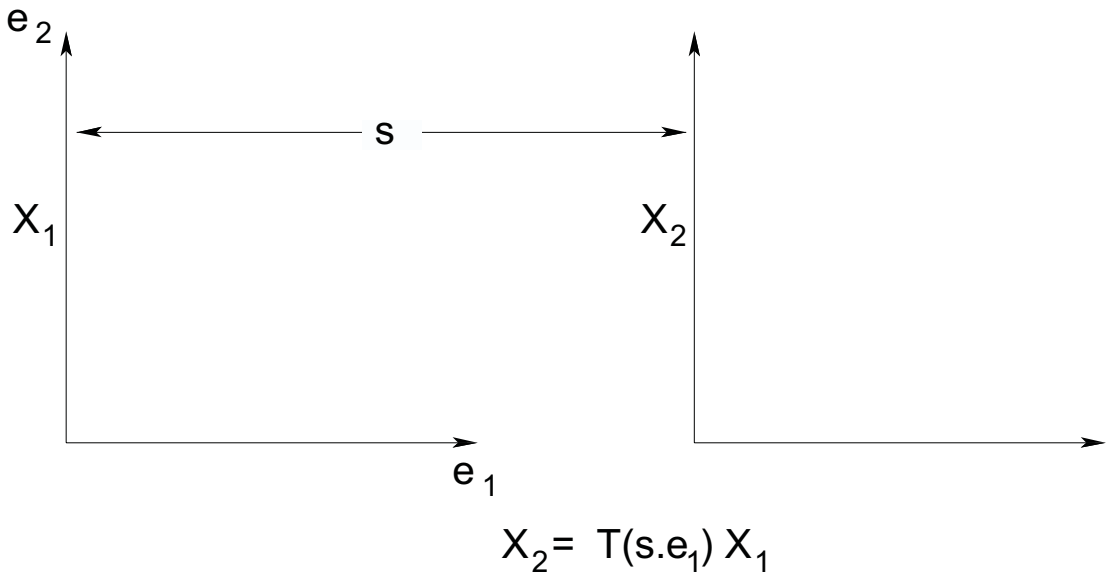

Figure 2.15: Example of the translation operation

$\mathbf{X}_2$ are also transformed by the same operator:

$$\mathbf{x}_2 = \mathcal{T}(\mathbf{v})\mathbf{x}_1 \tag{2.51}$$

Figure 2.15 shows an example of a translation operator in action.

The operator $\mathcal{R}(\theta, \mathbf{a})$ takes two arguments; a scalar value θ which is the amount of rotation in Radians, and a vector value $\mathbf{a}$ which must be a unit vector in the direction of the axis of rotation. The right hand rule is used to define the sign of the rotation angle. The notation:

$$\mathbf{X}_2 = \mathcal{R}(\theta, \mathbf{a})\mathbf{X}_1 \tag{2.52}$$

means that the reference frame $\mathbf{X}_2$ is obtained by rotating $\mathbf{X}_1$ by an angle θ about an axis along the unit vector $\mathbf{a}$ passing through the origin of $\mathbf{X}_1$. $\mathbf{a}$ is a relative vector, and may be measured either in $\mathbf{X}_1$ or $\mathbf{X}_2$, because $\mathbf{a}$ is the same when measured in either. Coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$ of a vector measured in $\mathbf{X}_1$ and $\mathbf{X}_2$ are also transformed by the same operator:

$$\mathbf{x}_2 = \mathcal{R}(\theta, \mathbf{a})\mathbf{x}_1 \tag{2.53}$$

Figure 2.16 shows the use of the rotation operator.

2.4.2 The Formate Face-Milling Process

Figure 2.17 shows the face-milling machine set up to cut gears using Formate process. The reference frame $\mathbf{X}_c$ is attached the cutter head. $\mathbf{X}_m$ is a reference frame attached to the machine, and $\mathbf{X}_w$ is a reference frame attached to the workpiece.

The machine has four settings. H is the horizontal setting, V is the vertical setting, Γ_m is the machine root angle, and X_p is the machine center to back setting.

The kinematics of this machine is completely described by the transformation operators that transform the cutter reference frame $\mathbf{X}_c$ to the workpiece reference frame $\mathbf{X}_w$.

The angle θ represents the rotation of the cutter head, and is called the cutter phase angle.

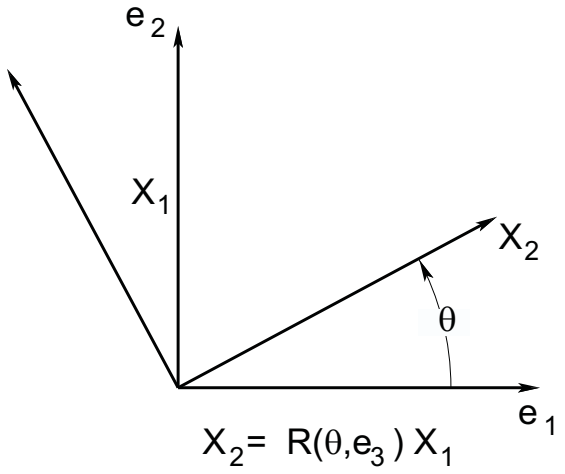

Figure 2.16: Example of the Rotation Operation

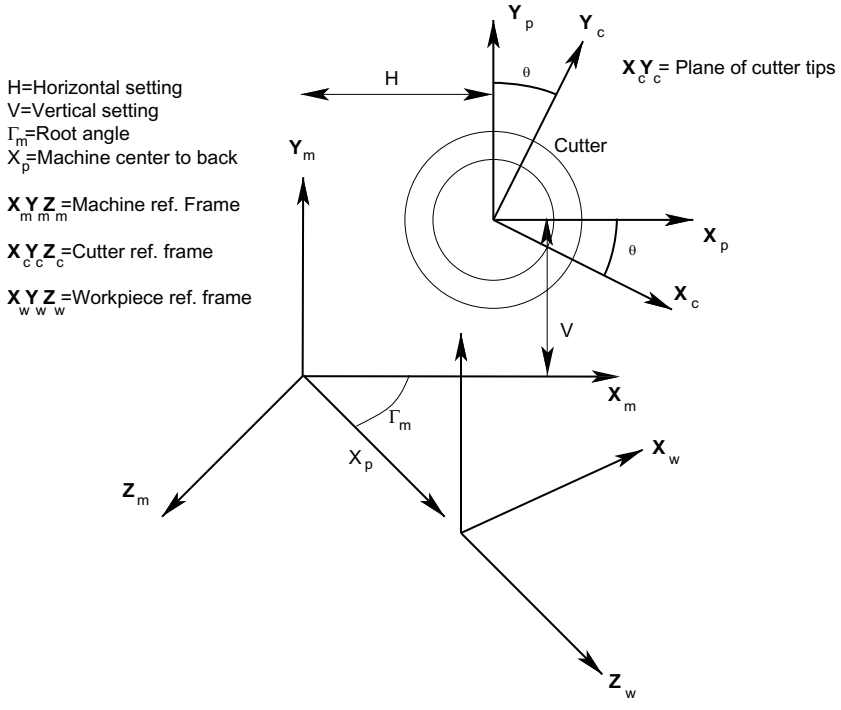

Figure 2.17: Machine settings and kinematics of the Formate process for face milled gears. The machine shown is set up for cutting a right handed gear.

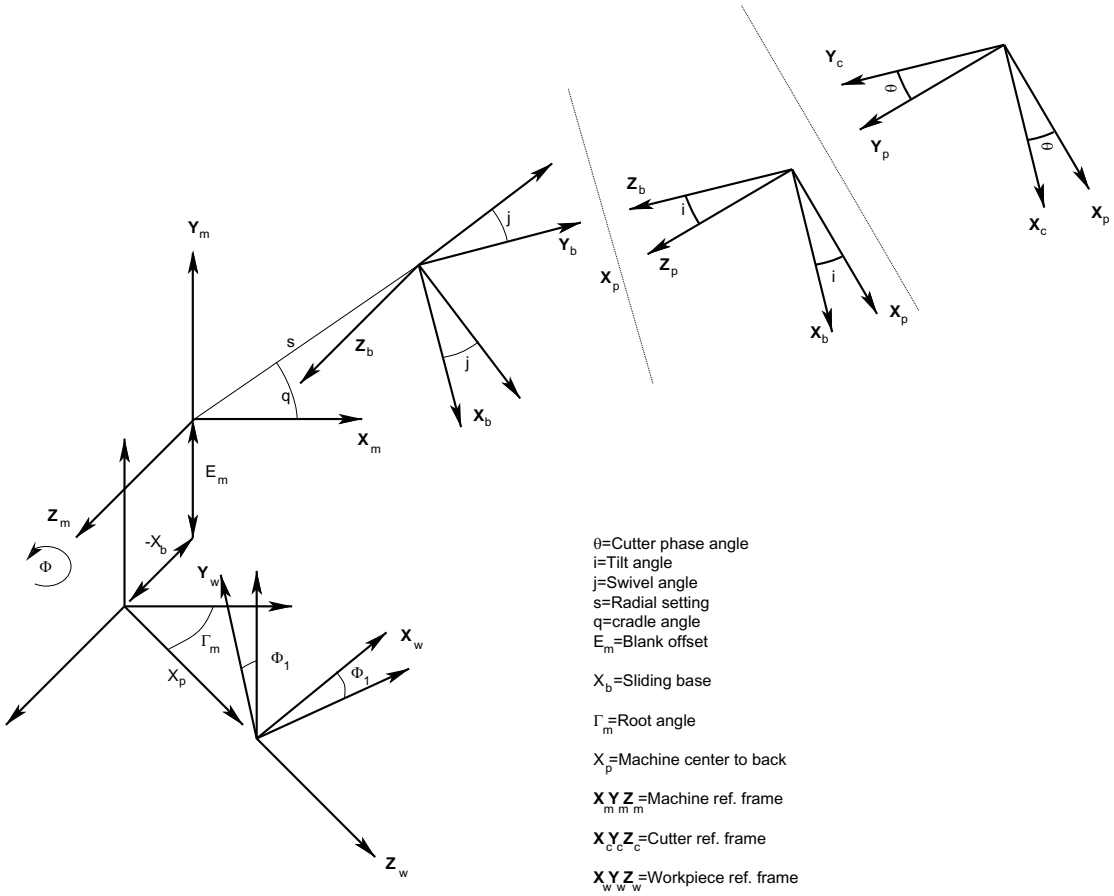

Figure 2.18: Machine settings and kinematics of the generation process for face milled gears The machine shown is set up for cutting a right handed pinion or gear.

For cutting a left handed gear,

$$\mathbf{X}_w = \mathcal{R}(\pi, \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2)\mathcal{T}(-H\mathbf{e}_1 + V\mathbf{e}_2)\mathcal{R}(\theta, \mathbf{e}_3)\mathbf{X}_c \tag{2.54}$$

and for a right handed gear

$$\mathbf{X}_w = \mathcal{R}(\pi, \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2)\mathcal{T}(-H\mathbf{e}_1 - V\mathbf{e}_2)\mathcal{R}(\theta, \mathbf{e}_3)\mathbf{X}_c \tag{2.55}$$

The surface of the gear is simply the surface swept by the cutting edges of the cutter blades in the workpiece reference frame.

2.4.3 The Generated Face-Milling Process

Figure 2.18 shows the face-milling machine set up to cut gears using the generated process. The reference frame $\mathbf{X}_c$ is attached the cutter head. $\mathbf{X}_m$ is a reference frame attached to the machine, and $\mathbf{X}_w$ is a reference frame attached to the workpiece.

There are several machine settings for the generation process. The angle i is called the tilt angle. The angle j is the swivel angle. s is the radial setting, The angle q is the cradle angle.

E_m is the blank offset. X_b is the sliding base setting. Γ_m is the machine root angle and X_p is the machine center to back setting.

The angle θ represents the rotation of the cutter head, and is called the cutter phase angle. The angle ϕ is the rotation of the cradle, and the angle ϕ_1 is the workpiece rotation.

For a left handed gear or pinion,

$$\begin{aligned}\mathbf{X}_w &= \mathcal{R}(\phi_1, \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2) \\ &\quad \mathcal{T}((X_b + H_1\phi + H_2\phi^2 + H_3\phi^3)\mathbf{e}_3) \\ &\quad \mathcal{T}(-(E_m + V_1\phi + V_2\phi^2 + V_3\phi^3)\mathbf{e}_2) \\ &\quad \mathcal{R}(-\phi, \mathbf{e}_3)\mathcal{R}(-\pi/2 + q, \mathbf{e}_3)\mathcal{T}(s\mathbf{e}_2) \\ &\quad \mathcal{R}(-j, \mathbf{e}_3)\mathcal{R}(-i, \mathbf{e}_2)\mathcal{R}(\theta, \mathbf{e}_3)\mathbf{X}_c \end{aligned} \tag{2.56}$$

where

$$\phi_1 = R_a(\phi - (C_2/2)\phi^2 - (C_3/6)\phi^3 - (C_4/24)\phi^4 - (C_5/120)\phi^5) \tag{2.57}$$

For a right handed gear or pinion

$$\begin{aligned}\mathbf{X}_w &= \mathcal{R}(\phi_1, \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2) \\ &\quad \mathcal{T}(-(X_b - H_1\phi - H_2\phi^2 - H_3\phi^3)\mathbf{e}_3) \\ &\quad \mathcal{T}((-E_m - V_1\phi - V_2\phi^2 - V_3\phi^3)\mathbf{e}_2) \\ &\quad \mathcal{R}(-\phi, \mathbf{e}_3)\mathcal{R}(\pi/2 - q, \mathbf{e}_3)\mathcal{T}(-s\mathbf{e}_2) \\ &\quad \mathcal{R}(j, \mathbf{e}_3)\mathcal{R}(-i, \mathbf{e}_2)\mathcal{R}(\theta, \mathbf{e}_3)\mathbf{X}_c \end{aligned} \tag{2.58}$$

where

$$\phi_1 = R_a(\phi - (-C_2/2)\phi^2 - (C_3/6)\phi^3 - (-C_4/24)\phi^4 - (C_5/120)\phi^5) \tag{2.59}$$

Equations 2.57 and 2.59 define the relation between the rolling motion ϕ of the cradle and the rotation ϕ_1 of the workpiece, for the left and right handed situations, respectively. R_a is the ratio of roll. C_2, C_3, C_4 and C_5 are the modified roll coefficients.

The settings H_1, H_2 and H_3 are 'helical motion' coefficients and V_1, V_2 and V_3 are vertical motion coefficients.

For each individual value of ϕ, the edges of the cutter blades sweep out a surface in the workpiece reference frame. As ϕ changes, the surface moves. The envelope of this moving surface is the surface of the generated pinion.

2.4.4 The Face-Hobbing Process Without Generating Motion

Creating a gear by the non-generated form of the face-hobbing process is very similar to the Formate form of the face-milling process. The same four machine settings are used. H is the horizontal setting, V is the vertical setting, Γ_m is the machine root angle, and X_p is the machine center to back setting.

For cutting a left handed gear,

$$\mathbf{X}_w = \mathcal{R}((N_c/N_g)\phi + \pi, \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2)\mathcal{T}(-H\mathbf{e}_1 + V\mathbf{e}_2)\mathcal{R}(\phi, \mathbf{e}_3)\mathbf{X}_c \tag{2.60}$$

and for a right handed gear

$$\mathbf{X}_w = \mathcal{R}((N_c/N_g)\phi + \pi, \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2)\mathcal{T}(-H\mathbf{e}_1 - V\mathbf{e}_2)\mathcal{R}(\phi, \mathbf{e}_3)\mathbf{X}_c \tag{2.61}$$

Here ϕ is the rotation of the cutter head. Because this is a face hobbing process, the rotation of the cutter head is synchronized with the rotation of the workpiece (the gear). The rotation of the workpiece is $(N_c/N_g)\phi$. N_c is the number of cutter blade groups on the cutter head and N_g is the number of teeth on the workpiece. This constant relationship between rotation angles ensures that exactly one blade group passes through each tooth space on the workpiece.

The surface of the gear is simply the surface that the cutter blade edge sweeps in the workpiece reference frame.

2.4.5 The Face-Hobbing Process With Generating Motion

The basic machine setup for the face-hobbing process is very similar to the generated face-milling process. The angle i is called the tilt angle. The angle j is the swivel angle. s is the radial setting. The angle q is the cradle angle. E_m is the blank offset. X_b is the sliding base setting. Γ_m is the machine root angle and X_p is the machine center to back setting.

While cutting the pinion by the face-hobbing process, generating motion is used in addition to the cutting motion. So there are two independent machine motions, represented by the motion parameters ϕ and ψ

For cutting a left handed pinion,

$$\begin{aligned}\mathbf{X}_w &= \mathcal{R}(\phi R_a + \psi(N_c/N_p), \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2)\mathcal{T}(X_b\mathbf{e}_3 - E_m\mathbf{e}_2) \\ &\quad \mathcal{R}(-\phi, \mathbf{e}_3)\mathcal{R}(-\pi/2 + q, \mathbf{e}_3)\mathcal{T}(s\mathbf{e}_2)\mathcal{R}(-j, \mathbf{e}_3)\mathcal{R}(-i, \mathbf{e}_2)\mathcal{R}(\psi, \mathbf{e}_3)\mathbf{X}_c \end{aligned} \quad (2.62)$$

and for a right handed pinion

$$\begin{aligned}\mathbf{X}_w &= \mathcal{R}(\phi R_a + \psi(N_c/N_p), \mathbf{e}_3)\mathcal{T}(X_p\mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma_m, \mathbf{e}_2)\mathcal{T}(X_b\mathbf{e}_3 + E_m\mathbf{e}_2) \\ &\quad \mathcal{R}(-\phi, \mathbf{e}_3)\mathcal{R}(\pi/2 - q, \mathbf{e}_3)\mathcal{T}(-s\mathbf{e}_2)\mathcal{R}(j, \mathbf{e}_3)\mathcal{R}(-i, \mathbf{e}_2)\mathcal{R}(\psi, \mathbf{e}_3)\mathbf{X}_c \end{aligned} \quad (2.63)$$

The parameter ψ represents the cutting motion. The workpiece has a rotation $\psi(N_c/N_p)$ that is synchronized with the cutting motion of the cutter head. Here N_c is the number of blade groups on the cutter head and N_p is the number of teeth in the pinion. This constant ratio ensures that exactly one blade group passes through each tooth space on the pinion.

A generating 'roll' is superimposed on top of the cutting motion. The motion parameter ϕ represents a rolling motion of the entire cradle. A corresponding synchronized rotation of ϕR_a is added to the pinion.

The cradle represents an imaginary 'crown' gear that is in mesh with pinion being generated. The cutting motion simulates the process of cutting the tooth form of the crown gear, and the rolling motion represents the pinion being generated as the conjugate of the crown gear.

For a fixed value of roll ϕ, the cutter blade sweeps out a surface in the pinion reference frame. As ϕ changes, the cradle and pinion move, and this swept surface also moves. The envelope of the family of all the swept surfaces is the surface of the generated pinion.

2.5 Straight Bevel Gears and the Octoid Form

Straight bevel gears are common in automotive differential mechanisms. Several generation techniques are used in the industry, and appear under different trade names, but they are all based on the spherical involute geometry [13, 14]. This spherical involute form is also referred to as the 'Octoid' form, and has many similarities to planar involutes. It is called the octoid form because the full path of contact theoretically extends over a curve that resembles the figure eight.

Just as a planar involute is generated by a rack, a spherical involute surface is generated by a crown rack,

Figure 2.19 shows the cutter blade with a blade angle α, placed in its reference frame $\mathbf{X}_c$. Cutting motion is obtained by using a cutting motion parameter t in the transformation from the cutter reference frame $\mathbf{X}_c$ to the cradle reference frame $\mathbf{X}_1$:

$$\mathbf{X}_1 = \mathcal{R}(-\pi/2, \mathbf{e}_3)\mathcal{T}(t(\mathbf{e}_2 + tan(\beta)\mathbf{e}_3 - tan(\beta)tan(\alpha)\mathbf{e}_1))\mathbf{X}_c \quad (2.64)$$

Here β is the dedendum angle of the workpiece.

As t varies, the cutting edge sweeps out the planar surface of the crown rack as observed from the cradle. and shown in Figure 2.20.

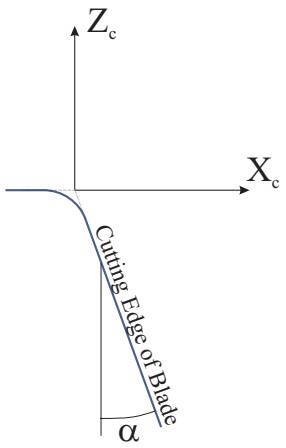

Figure 2.19: The cutter blade used for cutting a straight-bevel gear.

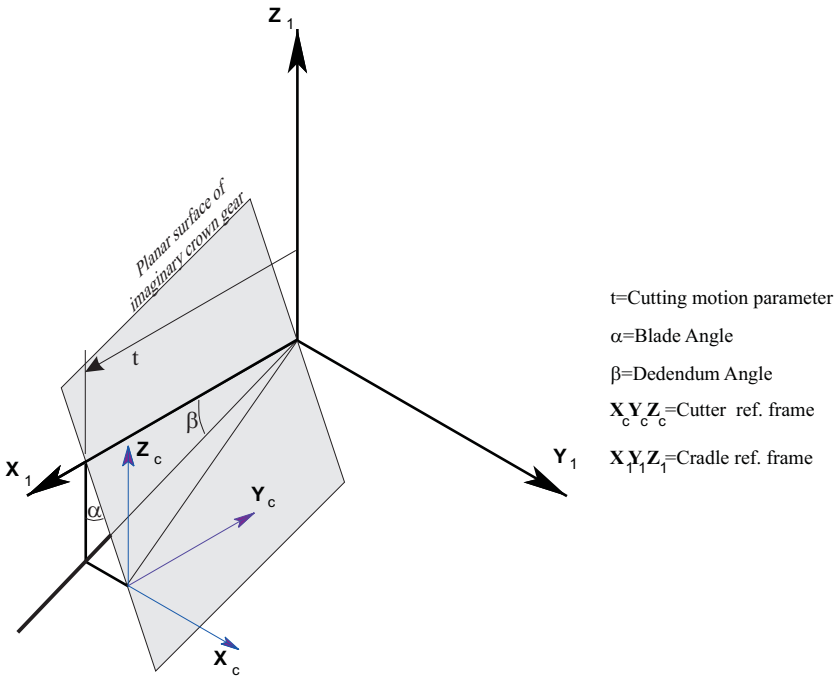

Figure 2.20: The cutting motion of the blade while cutting a straight-bevel gear.

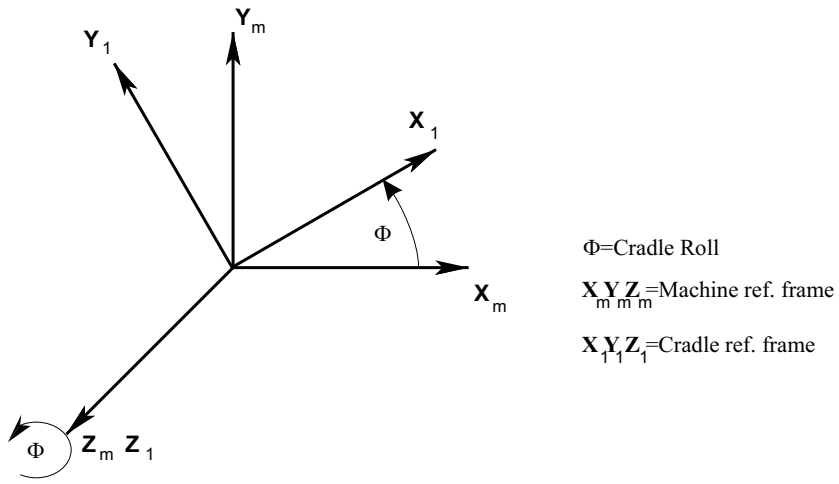

Figure 2.21: The motion of the cradle while cutting a straight-bevel gear.

Generation motion is obtained by rotating the entire cradle and the workpiece with respect to the machine. If ϕ is the rotation of the cradle reference frame with respect to the machine (Figure 2.21):

$$\mathbf{X}_m = \mathcal{R}(-\phi, \mathbf{e}_3)\mathbf{X}_1 \tag{2.65}$$

then the workpiece rotation is $\phi_1 = R_a\phi$, where R_a is the ratio of roll. If Γ is the pitch cone angle of the workpiece, then the transformation from the machine reference frame and the rolling workpiece reference frame is:

$$\mathbf{X}_w = \mathcal{R}(R_a\phi, \mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma, \mathbf{e}_2)\mathbf{X}_m \tag{2.66}$$

This workpiece rolling motion is depicted in Figure 2.22. The ratio of roll is calculate assuming that the crown rack moves like an imaginary bevel gear of pitch cone angle $\pi/2$

$$R_a = sin(\pi/2)/sin(\Gamma) = 1/sin(\Gamma) \tag{2.67}$$

The composite transformation from the cutter reference frame $\mathbf{X}_c$ to the workpiece reference frame $\mathbf{X}_w$ incorporates both the cutting motion and the generating motion:

$$\begin{aligned} \mathbf{X}_w \quad = \quad & \mathcal{R}(R_a\phi, \mathbf{e}_3)\mathcal{R}(\pi/2 - \Gamma, \mathbf{e}_2) \\ & \mathcal{R}(-\phi, \mathbf{e}_3) \\ & \mathcal{R}(-\pi/2, \mathbf{e}_3)\mathcal{T}(t(\mathbf{e}_2 + tan(\beta)\mathbf{e}_3 - tan(\beta)tan(\alpha)\mathbf{e}_1)) \\ & \mathbf{X}_c \end{aligned} \tag{2.68}$$

The finite element model of the straight bevel gear tooth surface is easily generated by tracing out the surface swept out by the cutter blade on the workpiece.

2.6 Hierarchical Structures

The finite element models of transmission components can be extremely large. This is especially so for the gears in the system. Each individual tooth of a gear can have several thousand degrees of freedom. There may be up to a hundred teeth on a gear, and there may be several gears mounted on a shaft. This whole assembly forms a single body. The total number of degrees of

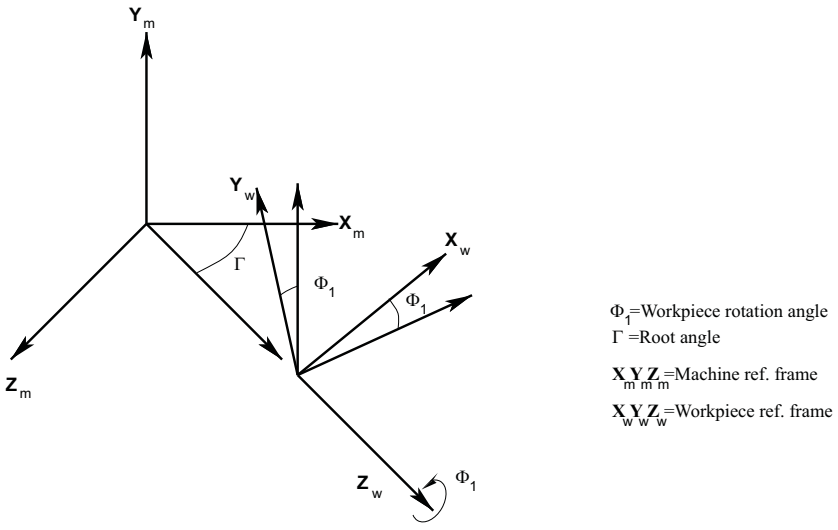

Figure 2.22: The motion of the workpiece while cutting a straight-bevel gear.

freedom can be of the order or several hundred thousand on one rotor, and the total number of degrees of freedom in the entire model can be in the millions.

In order to compute a compliance matrix needed by the contact equation solver, we need to apply a unit load at each contact grid cell and determine the response of the entire system to that unit load. So the total number of load cases that need to be solved for the system is equal to the number of grid cells. The number of grid cells can number in the several thousands. This implies that we would need to make several thousand backsubstitutions through a model that might have millions of degrees of freedom.

This process would need to be repeated for each time step of interest.

Using brute force to carry out this process is simply not feasible. In order to make the problem solvable, we need to extensively apply substructuring techniques, and we need to take advantage of the cyclic symmetry of gears.

If we consider just a simple model of a single gear with N teeth, the total number of degrees of freedom the in the model would be of the order of N. The time needed to decompose the stiffness matrix would be of order N^3.

Alternatively, we can build a substructure (or 'super-element') consisting of only one tooth. Using two copies of this single tooth substructure, we could build a two-tooth substructure. Using two copies of this, we could build a four-tooth substructure, or we could build a three-tooth substructure using one copy of the single tooth substructure and one copy of the two-tooth substructure. Repeating this process, it is possible to build a model of the entire gear using this hierarchy of substructures.

Figure 2.23 shows such a hierarchy for a fifteen tooth gear. It has a five level hierarchy, with the single tooth model at the lowest level, and the fifteen tooth model at the highest level. The number of substructures at any level can always be restricted to two or less. If that is done, then for an N tooth gear, the total number of substructures will be of the order of $log(N)$.

While building a substructure or superelement, only the degrees of freedom required at the next higher level are retained as master degrees of freedom. The internal slave degrees of freedom are condensed out. If we follow this practice, then the most computationally intensive substructure is the lowest level one-tooth substructure. It will need to be decomposed only once and its computational needs are independent of the number of teeth in the model. The computational expense of each higher level is roughly the same. So the total computational cost

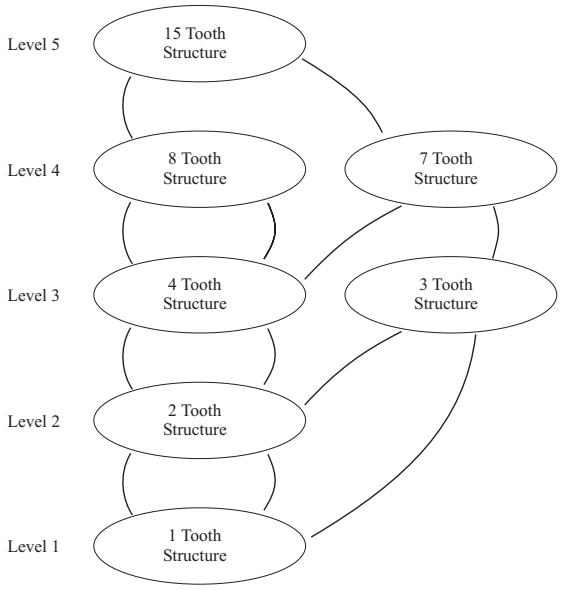

Figure 2.23: The hierarchy of substructures for a 15 tooth gear model.

of the entire gear N tooth gear model is of the order of $log(N)$. This represents a tremendous savings over the flat model that has a computational cost of the order of N^3.

All our gear system models use this technique.

The equations representing the behavior of a substructure at any level take the following form:

$$\mathbf{Kx} = \mathbf{f} + \mathbf{Ah} \tag{2.69}$$

$$\delta = \mathbf{Bx} + \mathbf{Ch} + \mathbf{d} \tag{2.70}$$

Here $\mathbf{K}$ is the stiffness matrix, $\mathbf{x}$ is a vector of the structure's degrees of freedom, $\mathbf{h}$ is a vector of internal contact forces, and $\mathbf{f}$ is a vector of external loads, including external contact forces. δ is a vector of the separations at the structure's internal contacts in the deformed condition. $\mathbf{d}$ is a vector of separation at the same internal contacts in the unloaded and undeformed state.

For a structure at the lowest level of the structure hierarchy, this set of equations is constructed using its finite element model. $\mathbf{K}$ is the stiffness matrix and $\mathbf{f}$ is the force vector. For a dynamic problem, the effective stiffness and force vectors obtained after time-discretization (Equation 2.12) are used instead.

$\mathbf{B}$ is obtained purely from geometric considerations. In most cases, $\mathbf{A} = \mathbf{B}^T$. $\mathbf{C} = 0$ at the lowest level.

For a higher level structure, the system of equations is obtained from the behavior of the lower level structures, as described below.

For a structure at any level of the hierarchy, we would like to partition the system 2.69 and condense out all internal degrees of freedom so that we arrive at a much smaller system of equations similar to 2.69. We partition $\mathbf{x}$ into slave degrees of freedom $\mathbf{x}_s$ and master degrees of freedom $\mathbf{x}_m$. The slave degrees of freedom are internal to the substructure, and do not connect with any other structure.

$$\mathbf{x} = \left\{ \begin{array}{c} \mathbf{x}_s \\ \mathbf{x}_m \end{array} \right\} \tag{2.71}$$

Accordingly partitioning 2.69,

$$\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}_{sm} \\ \mathbf{K}_{ms} & \mathbf{K}_{mm} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_s \\ \mathbf{f}_m \end{Bmatrix} + \begin{bmatrix} \mathbf{A}_s \\ \mathbf{A}_m \end{bmatrix} \mathbf{h} \tag{2.72}$$

From the upper partition of this equation we get:

$$\mathbf{K}_{ss}\mathbf{x}_s + \mathbf{K}_{sm}\mathbf{x}_m = \mathbf{f}_s + \mathbf{A}_s\mathbf{h} \tag{2.73}$$

or

$$\mathbf{x}_s = -\mathbf{K}_{ss}^{-1}\mathbf{K}_{sm}\mathbf{x}_m + \mathbf{K}_{ss}^{-1}\mathbf{f}_s + \mathbf{K}_{ss}^{-1}\mathbf{A}_s\mathbf{h} \tag{2.74}$$

which we can write as:

$$\mathbf{x}_s = \mathbf{D}\mathbf{x}_m + \mathbf{g}_s + \mathbf{P}\mathbf{h} \tag{2.75}$$

where we have defined

$$\mathbf{D} = -\mathbf{K}_{ss}^{-1}\mathbf{K}_{sm},\ \mathbf{g}_s = \mathbf{K}_{ss}^{-1}\mathbf{f}_s \text{ and } \mathbf{P} = \mathbf{K}_{ss}^{-1}\mathbf{A}_s \tag{2.76}$$

Equation 2.75 forms the 'dependence equation' for the substructure,

After the solution at the next higher level is complete, the value of $\mathbf{x}_m$ can be obtained from the higher level structure. Then 2.75 is used to compute the response $\mathbf{x}_s$ of the slave degrees of freedom. Computing $\mathbf{K}_{ss}^{-1}$ and $\mathbf{K}_{ss}^{-1}\mathbf{K}_{sm}$ is usually the most time consuming part of setting up the dependence equation.

Substituting for $\mathbf{x}_s$ from 2.75 into the lower partition of 2.72, we get

$$(\mathbf{K}_{mm} - \mathbf{K}_{ms}\mathbf{K}_{ss}^{-1}\mathbf{K}_{sm})\mathbf{x}_m = (\mathbf{f}_m - \mathbf{K}_{ms}\mathbf{K}_{ss}^{-1}\mathbf{f}_s) + (\mathbf{A}_m - \mathbf{K}_{ms}\mathbf{K}_{ss}^{-1}\mathbf{A}_s)\mathbf{h} \tag{2.77}$$

or

$$(\mathbf{K}_{mm} + \mathbf{D}^T\mathbf{K}_{sm})\mathbf{x}_m = (\mathbf{f}_m + \mathbf{D}^T\mathbf{f}_s) + (\mathbf{A}_m + \mathbf{D}^T\mathbf{A}_s)\mathbf{h} \tag{2.78}$$

We can rewrite this as

$$\mathbf{K}_{mm}^*\mathbf{x}_m = \mathbf{f}_m^* + \mathbf{A}_m^*\mathbf{h} \tag{2.79}$$

where

$$\mathbf{K}_{mm}^* = \mathbf{K}_{mm} + \mathbf{D}^T\mathbf{K}_{sm}\ ,\ \mathbf{f}_m^* = \mathbf{f}_m + \mathbf{D}^T\mathbf{f}_s \text{ and } \mathbf{A}_m^* = \mathbf{A}_m + \mathbf{D}^T\mathbf{A}_s \tag{2.80}$$

Similarly, if we partition the contact separation equation 2.70, we get

$$\begin{aligned} \delta &= \begin{bmatrix} \mathbf{B}_s & \mathbf{B}_m \end{bmatrix} \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{Bmatrix} + \mathbf{d} && (2.81)\\ &= \mathbf{B}_s\mathbf{x}_s + \mathbf{B}_m\mathbf{x}_m + \mathbf{d} && (2.82)\\ &= \mathbf{B}_s(\mathbf{D}\mathbf{x}_m + \mathbf{g}_s + \mathbf{P}\mathbf{h}) + \mathbf{B}_m\mathbf{x}_m + \mathbf{d} && (2.83)\\ &= (\mathbf{B}_m + \mathbf{B}_s\mathbf{D})\mathbf{x}_m + (\mathbf{B}_s\mathbf{P})\mathbf{h} + (\mathbf{B}_s\mathbf{g}_s + \mathbf{d}) && (2.84)\\ & && (2.85) \end{aligned}$$

or

$$\delta = \mathbf{B}^*\mathbf{x}_m + \mathbf{C}^*\mathbf{h} + \mathbf{d}^* \tag{2.86}$$

where

$$\mathbf{B}^* = \mathbf{B}_m + \mathbf{B}_s\mathbf{D}\ ,\ \mathbf{C}^* = \mathbf{B}_s\mathbf{P} \text{ and } \mathbf{d}^* = \mathbf{B}_s\mathbf{g}_s + \mathbf{d} \tag{2.87}$$

Gathering 2.79 and 2.86 together, we get

$$\begin{aligned} \mathbf{K}_{mm}^*\mathbf{x}_m &= \mathbf{f}_m^* + \mathbf{A}_m^*\mathbf{h} && (2.88)\\ \delta &= \mathbf{B}^*\mathbf{x}_m + \mathbf{C}^*\mathbf{h} + \mathbf{d}^* && (2.89) \end{aligned}$$

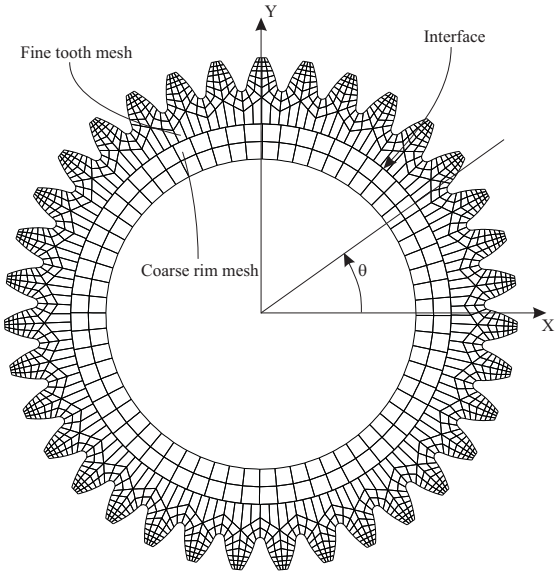

Figure 2.24: Interfacing two finite element models using Fourier degrees of freedom.

This system of equations is much smaller in size, but otherwise identical to the original system of equations 2.69 and 2.70.

We repeat this process for all the substructures at a particular level, and assemble all their individual equations 2.88 and 2.89 to construct the system of equations 2.69 and 2.70 for the next higher level. This process is repeated till we have the small system of equations at the highest level of the structure.

At the highest level, we proceed to actually solve 2.88 and 2.89 for $\mathbf{x}_m$ and $\mathbf{h}$. Once we have a value for $\mathbf{x}_m$, we can use the dependency relationship 2.75 to compute $\mathbf{x}_s$. Once we have both $\mathbf{x}_m$ and $\mathbf{x}_s$, we have the value of $\mathbf{x}$. We now have the complete response of the highest level structure.

From the response of the highest level structure, we can extract the value of the $\mathbf{x}_m$ and $\mathbf{h}$ for all the lower level substructures, and solve for the values of their $\mathbf{x}_s$. By recursively repeating the process, we travel down the substructure hierarchy obtaining the complete response for each level.

The solution process we have just described consists of a bottom-up traversal and a top-down traversal of the hierarchy. During the bottom-up traversal, we assemble and condense the substructure matrices. At the end of this first traversal, we have the highest level system equations, which we solve. Then we perform a top-down traversal of the hierarchy, backsubstituting and calculating the response of all the degrees of freedom that had been condensed out.

At the end of this two stage process, we have the entire response of the entire system to the external excitation. The time requirement for a gear with N teeth is of the order $logN$.

2.7 Fourier Surfaces

Figure 2.24 shows a typical gear finite element model. Computational cost considerations dictate that we cannot use a fine mesh everywhere. So we divide the gear into two finite element regions. A fine mesh is used to model the gear tooth region. A much coarser mesh is used to model the 'rim' region.

At the cylindrical surface that forms the interface between these two finite element meshes, their nodes do not coincide, and the elements might be of a different order. In order to join the two finite element models together at the interface, we expand the displacement field $\mathbf{u}(\theta, z)$ as a Fourier series in θ and a polynomial series in z:

$$\mathbf{u}(\theta, z) = \left\{ \begin{array}{l} u_r(\theta, z) \\ u_\theta(\theta, z) \\ u_z(\theta, z) \end{array} \right\} \tag{2.90}$$

$$\begin{aligned} u_r(\theta, z) \;=\; & \left(\begin{array}{ll} a_{r00} & +a_{r01}cos(\theta) + a_{r02}cos(2\theta) + a_{r03}cos(3\theta) + \cdots + a_{r0n}cos(n\theta) \\ & +b_{r01}sin(\theta) + b_{r02}sin(2\theta) + b_{r03}sin(3\theta) + \cdots + b_{r0n}sin(n\theta) \end{array} \right) \\ + \; z \; & \left(\begin{array}{ll} a_{r10} & +a_{r11}cos(\theta) + a_{r12}cos(2\theta) + a_{r13}cos(3\theta) + \cdots + a_{r1n}cos(n\theta) \\ & +b_{r11}sin(\theta) + b_{r12}sin(2\theta) + b_{r13}sin(3\theta) + \cdots + b_{r1n}sin(n\theta) \end{array} \right) \\ + \; z^2 \; & \left(\begin{array}{ll} a_{r20} & +a_{r21}cos(\theta) + a_{r22}cos(2\theta) + a_{r23}cos(3\theta) + \cdots + a_{r2n}cos(n\theta) \\ & +b_{r21}sin(\theta) + b_{r22}sin(2\theta) + b_{r23}sin(3\theta) + \cdots + b_{r2n}sin(n\theta) \end{array} \right) \\ + \; z^3 \; & \left(\begin{array}{ll} a_{r30} & +a_{r31}cos(\theta) + a_{r32}cos(2\theta) + a_{r33}cos(3\theta) + \cdots + a_{r3n}cos(n\theta) \\ & +b_{r31}sin(\theta) + b_{r32}sin(2\theta) + b_{r33}sin(3\theta) + \cdots + b_{r3n}sin(n\theta) \end{array} \right) \\ & \vdots \\ + \; z^m \; & \left(\begin{array}{ll} a_{rm0} & +a_{rm1}cos(\theta) + a_{rm2}cos(2\theta) + a_{rm3}cos(3\theta) + \cdots + a_{rmn}cos(n\theta) \\ & +b_{rm1}sin(\theta) + b_{rm2}sin(2\theta) + b_{rm3}sin(3\theta) + \cdots + b_{rmn}sin(n\theta) \end{array} \right) \end{aligned} \tag{2.91}$$

$$\begin{aligned} u_\theta(\theta, z) \;=\; & \left(\begin{array}{ll} a_{\theta00} & +a_{\theta01}cos(\theta) + a_{\theta02}cos(2\theta) + a_{\theta03}cos(3\theta) + \cdots + a_{\theta0n}cos(n\theta) \\ & +b_{\theta01}sin(\theta) + b_{\theta02}sin(2\theta) + b_{\theta03}sin(3\theta) + \cdots + b_{\theta0n}sin(n\theta) \end{array} \right) \\ + \; z \; & \left(\begin{array}{ll} a_{\theta10} & +a_{\theta11}cos(\theta) + a_{\theta12}cos(2\theta) + a_{\theta13}cos(3\theta) + \cdots + a_{\theta1n}cos(n\theta) \\ & +b_{\theta11}sin(\theta) + b_{\theta12}sin(2\theta) + b_{\theta13}sin(3\theta) + \cdots + b_{\theta1n}sin(n\theta) \end{array} \right) \\ + \; z^2 \; & \left(\begin{array}{ll} a_{\theta20} & +a_{\theta21}cos(\theta) + a_{\theta22}cos(2\theta) + a_{\theta23}cos(3\theta) + \cdots + a_{\theta2n}cos(n\theta) \\ & +b_{\theta21}sin(\theta) + b_{\theta22}sin(2\theta) + b_{\theta23}sin(3\theta) + \cdots + b_{\theta2n}sin(n\theta) \end{array} \right) \\ + \; z^3 \; & \left(\begin{array}{ll} a_{\theta30} & +a_{\theta31}cos(\theta) + a_{\theta32}cos(2\theta) + a_{\theta33}cos(3\theta) + \cdots + a_{\theta3n}cos(n\theta) \\ & +b_{\theta31}sin(\theta) + b_{\theta32}sin(2\theta) + b_{\theta33}sin(3\theta) + \cdots + b_{\theta3n}sin(n\theta) \end{array} \right) \\ & \vdots \\ + \; z^m \; & \left(\begin{array}{ll} a_{\theta m0} & +a_{\theta m1}cos(\theta) + a_{\theta m2}cos(2\theta) + a_{\theta m3}cos(3\theta) + \cdots + a_{\theta mn}cos(n\theta) \\ & +b_{\theta m1}sin(\theta) + b_{\theta m2}sin(2\theta) + b_{\theta m3}sin(3\theta) + \cdots + b_{\theta mn}sin(n\theta) \end{array} \right) \end{aligned} \tag{2.92}$$

$$
\begin{aligned}
u_z(\theta,z) \;=\; & \left(\begin{array}{ll} a_{z00} & +a_{z01}cos(\theta)+a_{z02}cos(2\theta)+a_{z03}cos(3\theta)+\cdots+a_{z0n}cos(n\theta) \\ & +b_{z01}sin(\theta)+b_{z02}sin(2\theta)+b_{z03}sin(3\theta)+\cdots+b_{z0n}sin(n\theta) \end{array}\right) \\
+ \; z \; & \left(\begin{array}{ll} a_{z10} & +a_{z11}cos(\theta)+a_{z12}cos(2\theta)+a_{z13}cos(3\theta)+\cdots+a_{z1n}cos(n\theta) \\ & +b_{z11}sin(\theta)+b_{z12}sin(2\theta)+b_{z13}sin(3\theta)+\cdots+b_{z1n}sin(n\theta) \end{array}\right) \\
+ \; z^2 \; & \left(\begin{array}{ll} a_{z20} & +a_{z21}cos(\theta)+a_{z22}cos(2\theta)+a_{z23}cos(3\theta)+\cdots+a_{z2n}cos(n\theta) \\ & +b_{z21}sin(\theta)+b_{z22}sin(2\theta)+b_{z23}sin(3\theta)+\cdots+b_{z2n}sin(n\theta) \end{array}\right) \\
+ \; z^3 \; & \left(\begin{array}{ll} a_{z30} & +a_{z31}cos(\theta)+a_{z32}cos(2\theta)+a_{z33}cos(3\theta)+\cdots+a_{z3n}cos(n\theta) \\ & +b_{z31}sin(\theta)+b_{z32}sin(2\theta)+b_{z33}sin(3\theta)+\cdots+b_{z3n}sin(n\theta) \end{array}\right) \\
& \vdots \\
+ \; z^m \; & \left(\begin{array}{ll} a_{zm0} & +a_{zm1}cos(\theta)+a_{zm2}cos(2\theta)+a_{zm3}cos(3\theta)+\cdots+a_{zmn}cos(n\theta) \\ & +b_{zm1}sin(\theta)+b_{zm2}sin(2\theta)+b_{zm3}sin(3\theta)+\cdots+b_{zmn}sin(n\theta) \end{array}\right)
\end{aligned}
\tag{2.93}
$$

The terms $a_{.ij}$ and $b_{.ij}$ in 2.91, 2.92 and 2.93 are the coefficients of this expansion. We treat these 'Fourier degrees of freedom' as the master degrees of freedom at the interface. The nodal degrees of freedom of all the nodes that lie on this cylinder are expressed in terms of these master degrees of freedom using this Fourier expansion.

By using the same master degrees of freedom on the two sides of the interface, we automatically 'join' the two models together.

Of these coefficients, some terms correspond to rigid body motions of the interface surface. For example, a non-zero value for a_{r01} corresponds to a translation in the X direction. b_{r01} corresponds to rigid body motion in the Y direction and a_{z00} corresponds to rigid body motion in the Z direction. a_{r11} corresponds to a rotation about the Y axis. b_{r11} corresponds to a rotation about the X axis. Finally, $a_{\theta 00}$ corresponds to a rotation about the Z axis.

Sometimes we need to enforce constraints only on the rigid body components of deformation at a Fourier interface, and allow all non-rigid body modes to deform freely. In that case, we only constrain a_{r01}, b_{r01}, a_{z00}, a_{r11}, b_{r11} and $a_{\theta 00}$ to zero.

2.8 Bearing Contact Models

One of our major goals was to incorporate a detailed contact model of rolling element bearings in our gear and transmission models. The contact between the rolling elements in a bearing and the inner and outer races is very similar to gears. So only minor modifications were needed to the contact analysis program *Calyx*. The changes were related to the fact that the contact surface on a roller or ball wraps all the way around the ball, and is closed, unlike the contact surface on a gear tooth which is open.

Figure 2.25 shows a schematic diagram of our rolling element bearing implementation. By varying the contact angle and the crown curvature on the roller and on the inner and outer races, we can build a cylindrical roller bearing (Figure 2.26), a tapered roller bearing (Figure 2.27), or even a ball bearing (Figure 2.28). The clearance at the rollers can be controlled by the user. A negative clearance can be used to introduce pre-load in the bearings.

We have made this detailed bearing implementation a part of the *Planetary3D* and *Transmission3D* software packages. Further details are provided in the user's manuals.

2.9 Transmission Noise Radiation

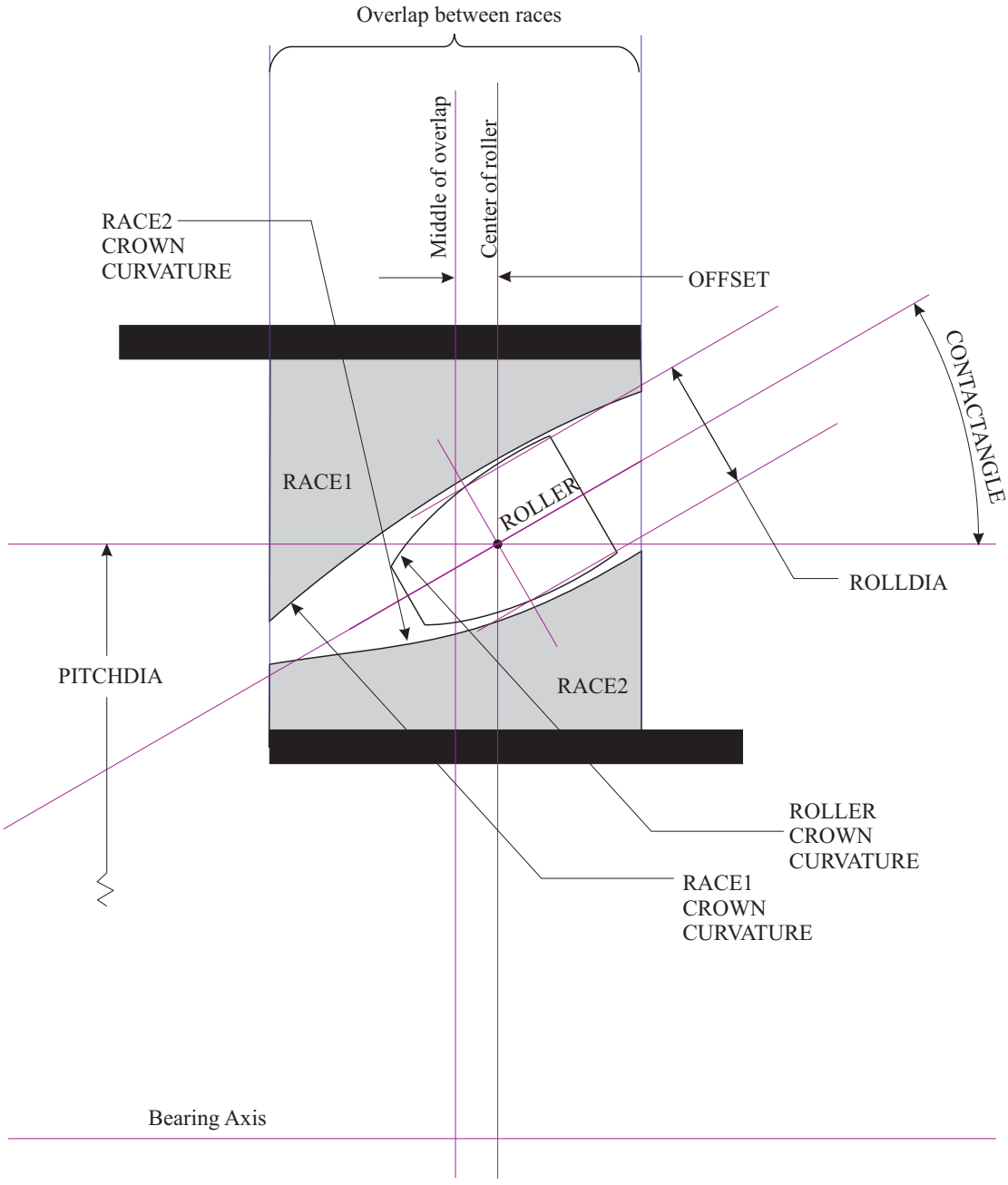

Figure 2.25: Roller bearing implementation

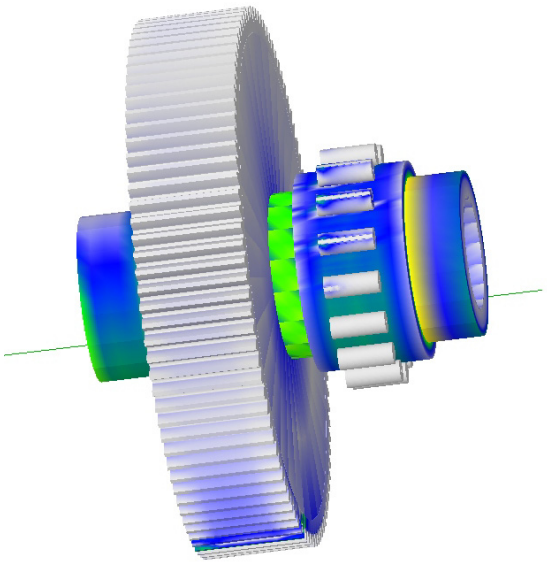

Figure 2.26: Cylindrical roller bearing.

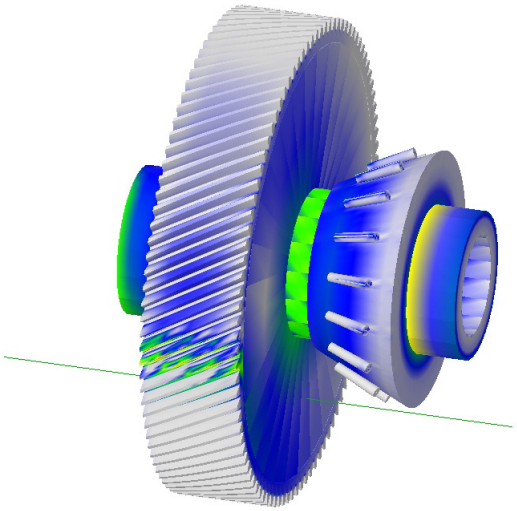

Figure 2.27: Tapered roller bearing.

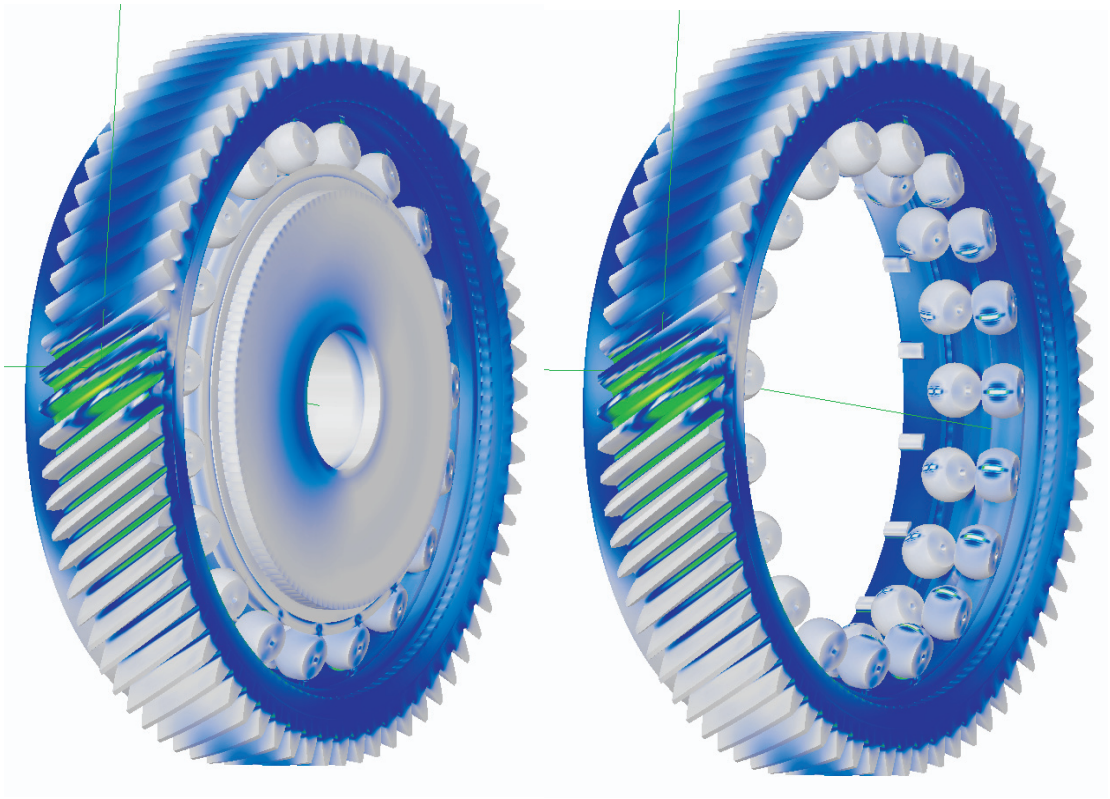

Figure 2.28: A helical gear on a double row ball bearing.

In order to predict the acoustic noise radiated by a transmission housing, we implemented a acoustic boundary element computer program based on the direct boundary element method. The boundary element program was put through a series of validation tests to compare its predictions with closed form solutions. The details are provided in the housing noise radiation theory manual [15].

The next step was to demonstrate how this capability will be used to compute the noise radiation from vibrating gearbox housings. This functionality was demonstrated using the example of the gearbox from a gear noise test rig developed at NASA Lewis Research Center [17].

We went through the following process for the gearbox:

- A finite element model of the gearbox housing was built from the geometry model. The computed structural mode shapes and natural frequencies were computed in *COSMIC NASTRAN*, and were validated against the published experimental modal analysis [17].
- Using assumed time domain bearing loads, we excited the structural model of the housing. In the future, we intend to use the output from static transmission housing models similar to the ones described in chapters 7 and 8 to provide this time domain excitation. The time domain loads were transformed into the frequency domain to obtain the load amplitudes as a function of frequency. At each frequency step, the housing structural response was computed using *COSMIC NASTRAN* via modal superposition.
- Results from the structural analysis (mode shapes or forced response) were converted into IDEAS universal format (dataset 55 - data at nodes).
- The structural results (in IDEAS universal dataset format) were imported into our BEM acoustics program using an interface developed for this purpose. The structure motion was imposed as a velocity boundary condition for the acoustic model.
- The BEM Acoustics program predicted the radiated sound power, pressure levels on any field point mesh, radiation efficiency at each frequency.

The details of this exercise are provided in the Housing Noise Radiation Testing manual [16]. We consider the development of the noise radiation prediction capability as an important success of our SBIR Phase II effort.

2.10 Multiple CPU structural and Contact Solver

We see the use of multiple CPUs as a potential means of accelerating the performance of *Calyx* in the future. Towards this end, we developed a version of *Calyx* that can take advantage of multiple processors on a shared memory architecture. An example of a multiple processor machine with shared memory is the SGI Origin 2000. Even desktop computers with multiple Intel CPUs running under Windows or Linux can be used.

We wrote special routines to farm out computational tasks to various CPUs, to synchronize their execution, and to collect the results. On Windows machines, we used the Win32 Threads API to accomplish this. On the SGI and Linux machines, we used the POSIX threads (pthread) library to accomplish this task.

We have a prototype of the multithreaded *Calyx* running now on all three platforms.

Chapter 3

Software Architecture

The software developed here consists of several packages. The *Planetary2D* package is used for the static and dynamic analysis of two-dimensional models of simple planetary systems. *Helical3D* is used for the three-dimensional analysis of external and internal spur and helical gear pairs. The *HypoidFaceMilled* package is used for the three-dimensional analysis of spiral-bevel and hypoid gear pairs generated by the face-milling method. *Planetary3D* is for the analysis of multiple mesh systems consisting of spur and helical external and internal gear pairs on parallel shafts, with an arbitrary system of planetary carriers and pinions, housings and shafts. *Transmission3D* is the package that encompasses the features of all the other packages, and also allows non-parallel axis systems, with bevel and hypoid gears.

Each of these packages comprises of the software modules shown in Figure 3.1.

Calyx is the contact analysis code. It communicates with the outside world through a programming language.

Multyx is capable of communicating with the user through an easy to use menu-based interface. It translates the user's commands into the appropriate programming language statements and sends them on to *Calyx*.

In addition to the user interface, Multyx also has built-in model generators. The models described here are all generated by *Multyx*. It also has post-processing and data extraction code, to help manipulate the results of analysis from *Calyx*.

Multyx and *Calyx* are designed as portable code, and can run on any system that supports standard *C++*. For the sake of portability, *Multyx*'s menu system is command line based, and does not use any of the GUI features such as buttons, windows or mouse interaction. The following dialog shows some of the command line interface of *Multyx*.

```
E:>multyx
MultyX v.1.06, Copyright Advanced Numerical Solutions Dec 21 2000
MultyX>post ok patt
MultyX.PostProc.1/11.Pattern>HELP
MENU            Show menu
?               Show menu
HELP            Show menu
EXIT            Return to main menu.
QUIT            Return to main menu.
START           Draw the contact pattern.
CLEAR           Clear the graphics page.
SURFACEPAIR     Surface pair (Currently=PINION1_SURFACE1_SUN_SURFACE1_)
MEMBER          Member (Currently=PINION1)
TOOTHBEGIN      20              Tooth no. or instance no. of surface.
```

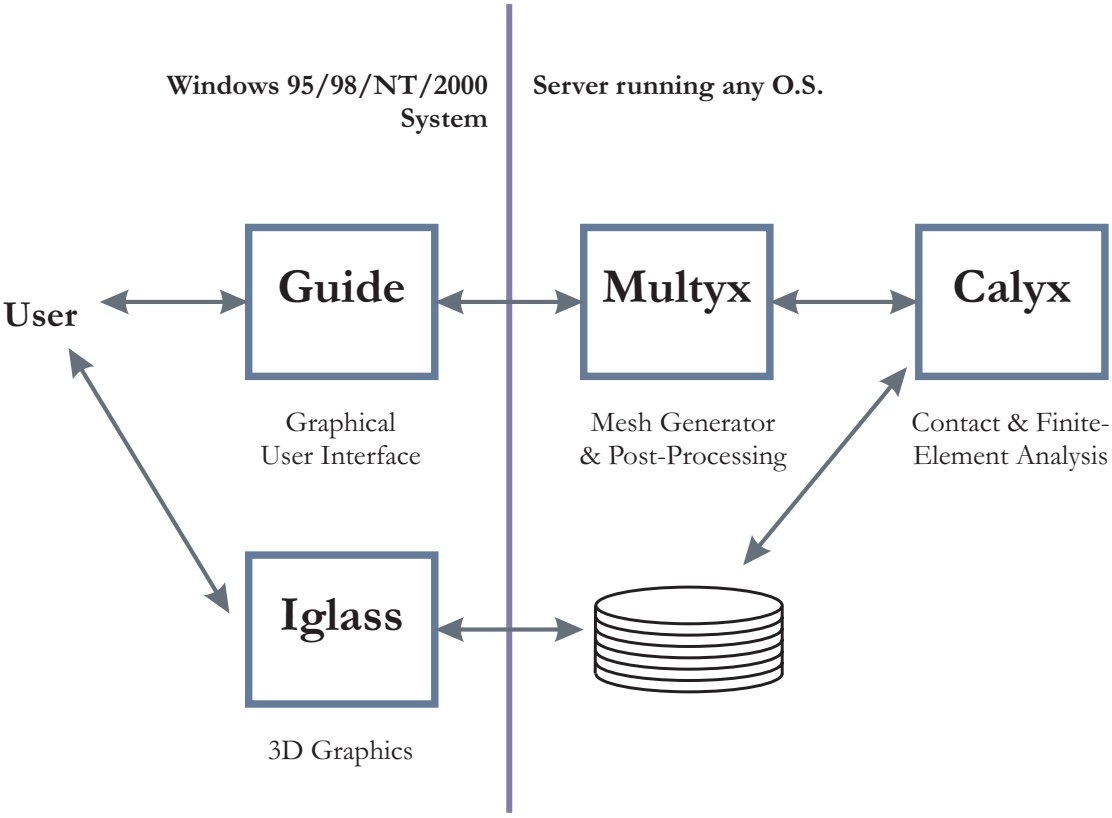

Figure 3.1: The computer programs in a software package based on *Calyx*

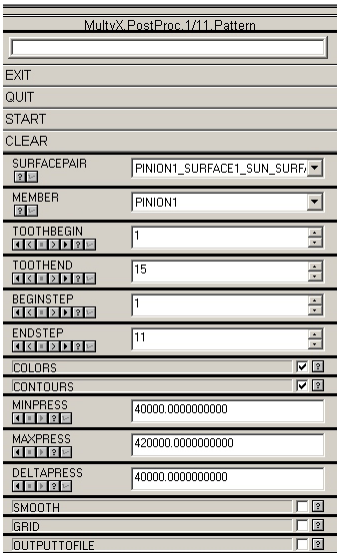

Figure 3.2: The menu presented to the user by Guide

```
TOOTHEND        2               Tooth no. or instance no. of surface.
BEGINSTEP       1               Time/Roll angle step at which to begin search.
ENDSTEP         11              Time/Roll angle step at which to end search.
COLORS          Whether to render the model in color (Enabled)
CONTOURS        Whether to draw pressure contours (Enabled)
MINPRESS        4.000000E+004   Level of lowest press. contour.
MAXPRESS        4.200000E+005   Level of highest press. contour.
DELTAPRESS      4.000000E+004   Spacing between press. contours.
SMOOTH          (FALSE)         Whether to smooth the pressure contours.
OUTPUTTOFILE    Whether to write data to file. (Disabled)
MultyX.PostProc.1/20.Pattern>START
```

Guide is a program that provides a Graphical User Interface (GUI) to *Multyx*. *Guide* translates each of *Multyx*'s dialogs and presents them to the user in a graphical form. The command line menu described above is presented to the user as shown in Figure 3.2.

In addition, *Guide* provide the user with convenient ways of viewing the graphics, and helps the user convert the graphics into Microsoft formats and into Encapsulated PostScript (EPS) files.

Although *Guide* enhances the friendliness of *Multyx*, it is not required. All the features of *Multyx* can be accessed without *Guide*. The connection between *Guide* and Multyx is based on the TCP/IP telnet protocol when they are running on different computers. When running on the same computer, they communicate through named pipes. *Guide* is a heavy user of advanced operating system features including GUI support, multi-threading support,

3.1 Calyx

Calyx is the program that actually carries out all analysis, pre- and post-processing computations. It can be used in stand-alone mode, or controlled through multyx. In depth details about *Calyx* can be found in its user's manual [18], but an overview of its salient features is provided here.

3.1.1 The Calyx Programming Language

The programming language through which *Calyx* receives its input looks very similar to *C++*. *Calyx* uses an interpreter instead of a compiler.

The *Calyx* language is not a strongly typed language. All variables are of the same type, and are declared as `var`s. This makes the language simpler to implement.

Three different data types have been implemented: a numerical data type, boolean data type, and a string data type. Data structures are formed by aggregating data items of different types in arrays. The arrays can, in turn, be used as parts of other arrays.

The *Calyx* programming language is designed to allow for building and manipulating symbolic expressions. This is important because it enables *Calyx* to compute velocities and accelerations from symbolic descriptions of the systems kinematics.

Loops are programmed using the `for`, `while` and `do-while` constructs. An `if` and `if-else` construct is available for conditional execution. A `switch` statement similar to that of *C++* can be used for more complicated switching.

Functions (subroutines) can be built by the user using the `function` declaration. Function parameters can be declared as `in`, `out`, or `inout` depending of whether the function parameter is for input only, output only, or for both. Unlike *C++*, local functions are allowed. So we can declare functions within other function declarations.

Variable names may be of arbitrary length. Variable and function scopes are limited to the statement block within which they are declared.

Exceptions can be thrown using the `throw` statement and caught using `try` and `catch` sections.

The grammar of the language is shown in Tables tables 3.1 and 3.2.

3.1.2 Setting up the System Kinematics

Calyx uses the two basic functions `Translate()` and `Rotate()` to build symbolic transformation matrices or operators for translational and rotational motion. These operators can be built and compounded in a myriad of ways, allowing us to describe the kinematics of almost all mechanical systems.

The `Translate(v)` function takes one argument; a translation vector `v`. The effect of multiplying a vector by this operator is to transform the coordinates of the vector to a new reference frame that has translated by an amount `v` from its original position.

The function `Rotate(theta,a)` takes two arguments; a scalar value `theta` which is the amount of rotation in Radians, and a vector value `a` which is a unit vector in the direction of the axis of rotation. The right hand rule is used to define the sign of the rotation angle `theta`. The effect of multiplying a vector by this operator is to transform the coordinates of the vector to a new reference frame that has rotated by an amount `theta` about the axis pointing along `a`, from its original position.

Consider the example of specifying the kinematics of a planet in a planetary system model. Figure 3.3 shows the location of the planet. There are pin-position errors and runout errors that need to be taken into account when positioning the planet's pitch circle at arbitrary time t.

The following snippet of code shows how this is done with *Calyx*.

```
var Omega_ Planet =3300;
var Cent_Dist_Planet_Sun=10.0;
var Planet_1_Runout_Error:=0.000002*e1+0.000001*e2;
var Planet_1_Pin_Posn_Error:=0.000001*e1+0.000002*e2;
var XPlanet_1:=
Translate(Planet_1_Runout_Error)*
Rotate((Omega_Planet-Omega_Carrier)*Time+
```

Table 3.1: Calyx Language Grammar

Non-Terminal Symbol		Expansion
ϵ	$\rightarrow$	(empty)
input	$\rightarrow$	ϵ stmt_list_toplevel
stmt_list_toplevel	$\rightarrow$	ϵ
	|	stmt_list_toplevel statement
compound_stmt	$\rightarrow$	{ stmt_list }
stmt_list	$\rightarrow$	ϵ
	|	stmt_list statement
statement	$\rightarrow$	assign_decl_exp ;
	|	compound_stmt
	|	function_defn
	|	`if` (exp)statement
	|	`if` (exp)statement else statement
	|	`break` ;
	|	`continue` ;
	|	`return` ;
	|	`return` exp ;
	|	`for` (assign_decl_exp ; exp ; assign_exp) statement
	|	`while` (exp) statement
	|	`do` statement `while` (exp) ;
	|	`switch` (exp) compound_stmt
	|	`case` exp : statement
	|	`goto` IDENTIFIER ;
	|	IDENTIFIER : statement
	|	`default` : statement
	|	`throw` ;
	|	`throw` exp ;
	|	`try` statement `catch` () statement
	|	`try` statement `catch` (IDENTIFIER) statement
function_defn	$\rightarrow$	`function` IDENTIFIER (parameter_list) statement
parameter_list	$\rightarrow$	ϵ
	|	parameter_list_1
parameter_list_1	$\rightarrow$	parameter
	|	parameter_list_1 , parameter
parameter	$\rightarrow$	`in` IDENTIFIER
	|	`out` IDENTIFIER
	|	inout IDENTIFIER
assign_decl_exp	$\rightarrow$	assign_exp
	|	decl_list
decl_list	$\rightarrow$	var identdecl_list
identdecl_list	$\rightarrow$	identdecl_list , identdecl
	|	identdecl

Table 3.2: Calyx Language Grammar (contd.)

Non-Terminal Symbol		Expansion
identdecl →		IDENTIFIER
	|	IDENTIFIER = exp
	|	IDENTIFIER := exp
assign_exp →		ϵ
	|	addr = exp
	|	addr := exp
	|	exp
addr →		addr [exp]
	|	IDENTIFIER
exp →		or_list
or_list →		or_list `|` and_list
	|	and_list
and_list →		and_list `&` binary
	|	binary
binary →		binary relop binary
	|	binary `+` binary
	|	binary `-` binary
	|	binary `*` binary
	|	binary `/` binary
	|	binary `^` binary
	|	unary
relop →		`<`
	|	`<=`
	|	`>`
	|	`>=`
	|	`==`
	|	`!=`
unary →		addr
	|	aoterm
	|	`[` list `]`
	|	`+` unary
	|	`-` unary
	|	`!` unary
	|	floating_point_constant
	|	string_constant
	|	`TRUE`
	|	`FALSE`
aoterm →		aoterm `[` exp `]`
	|	IDENTIFIER (list)
	|	`(` exp `)`
list →		ϵ
	|	list_1
list_1 →		list_1 ; exp
	|	exp

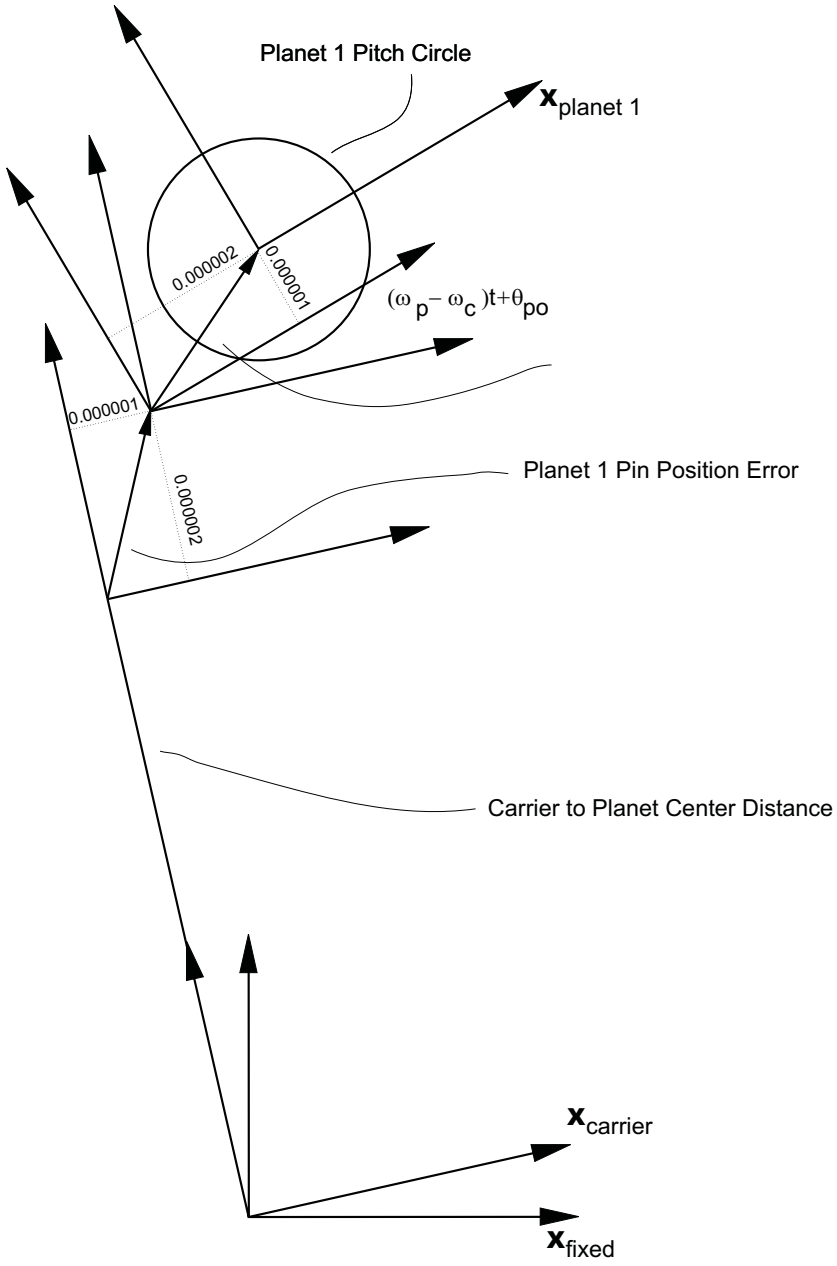

Figure 3.3: Planet reference frame definition

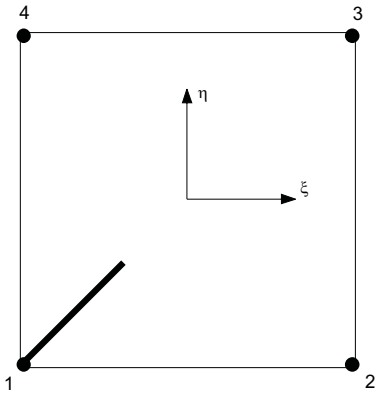

Figure 3.4: Type I coordinate connectivity

```
  Planet_1_Initial_Rotn,e3)*
Translate(Cent_Dist_Planet_Sun*e2+
          Planet_1_Pin_Posn_Error)*
XCarrier;
```

Here the reference frame for Planet 1, `XPlanet_1` is obtained by taking the Carrier reference frame `XCarrier`, and first applying a translation equal to the center distance between the planet and carrier in the Y direction ($= \mathbf{e}_2$) of the carrier reference frame, and the pin position error. Then a rotation is applied, and finally a translation is applied to account for the runout of the planet. The pin position error and the runout errors are defined as relative vectors. The pin position error vector is measured in the carrier reference frame `XCarrier`, and the runout error vector is measured in the planet reference frame `XPlanet_1`. The transformation operators are retained in symbolic form by *Calyx* so that it may later compute velocities and accelerations by differentiating these operators with respect to time.

Arbitrarily complex kinematics can be represented in this manner.

3.1.3 The Two-Dimensional Finite Element Library

Two different kinds of coordinate interpolation (or connectivity) have been implemented for interpolating coordinates in two-dimensional elements.

Elements that share part of the contacting surfaces have a very high coordinate accuracy requirement. Elements in the interior of the teeth need not have the same degree of coordinate accuracy.

Type I connectivity may be used for the elements requiring low coordinate accuracy. Figure 3.4 shows an element with type I connectivity. The element consists of only four coordinate nodes. Any side of a type I element may be shared with any side of another type I element.

Elements along the active profile and fillet region use type II connectivity. This kind of element has a variable number of coordinate nodes. This number n of coordinate nodes is even and greater than six, as shown in Figure 3.5. Nodes number $1, 2, 3, 5, 7, \ldots, n-1$ are all real coordinate nodes. Coordinate nodes number $4, 6, 8, 10, \ldots, n$ are virtual coordinate nodes used to define the variation of the surface normal along the $\eta = +1$ side of the element. This kind of element should not be interfaced with any other element along this ($\eta = +1$) side. The element may interface with type II elements on the $\xi = +1$ and $\xi = -1$ sides. Along the side $\eta = -1$, the element may interface with any side of a type I element, or with the $\eta = -1$ side of another type II element.

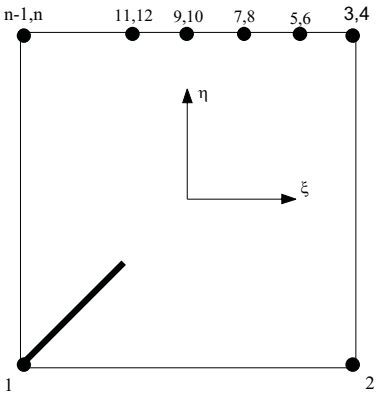

Figure 3.5: Type II coordinate connectivity

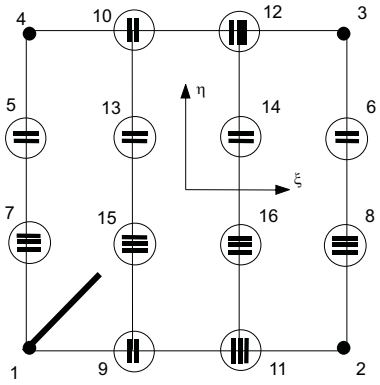

Figure 3.6: Displacement connectivity

At this time, only one kind of displacement interpolation or connectivity is available for two-dimensional finite elements. But this displacement connectivity has optional nodes, so that the displacement connectivity can be varied from a linear element connectivity all the way to a complete cubic element connectivity. Figure 3.6 shows an element with sixteen displacement nodes. The four corner nodes 1 to 4 are real displacement nodes. They are mandatory, and must be included. The remaining nodes are circled, indicating that they are optional. Any of these can be dropped from the element at the user's discretion. The number of dash marks within the circle indicates the order of interpolation that the node is associated with, in the direction of the line that the circle is drawn on. This figure is only a schematic representation. The nodes that are drawn on the boundaries of an element must be shared with elements that share the same boundary.

3.1.4 The Three-Dimensional Finite Element Library

Calyx supports hexahedral, pentahedral and tetrahedral elements.

Unlike other finite element codes, *Calyx* treats shell elements as a special form of a three-dimensional element in which the dimension along the ζ local axis is very small. Other codes treat shells as special two-dimensional element. The advantage of our treatment is that we can

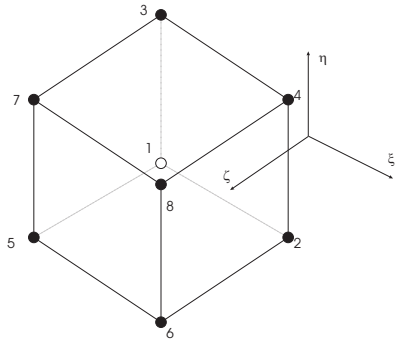

Figure 3.7: Linear Lagrangian hexahedral coordinate element

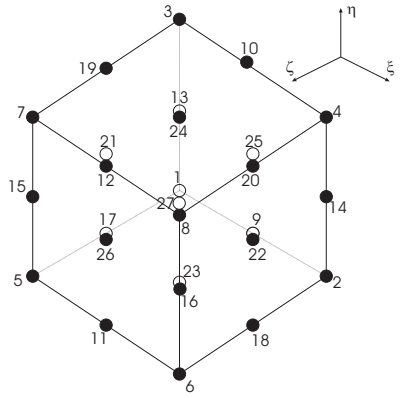

Figure 3.8: Quadratic Lagrangian hexahedral coordinate element

use any coordinate interpolation technique for both solid and shell elements. Thus we can have a highly detailed geometry for a shell element that can be used in contact. The hexahedral and pentahedral elements may be either solid or shell elements. The tetrahedral element cannot be used for shells.

For the interpolation of coordinates in hexahedral elements, we can use linear Lagrangian (Figure 3.7), quadratic Lagrangian (Figure 3.8), cubic Lagrangian, Linear FQP (Figure 3.9, see Vijayakar [19] for details on the FQP element), Quadratic FQP (Figure 3.10), cubic FQP (Figure 3.11) and special surface FQP interpolation (Figure 3.12). The surface FQP coordinate interpolation is used three-dimensional gear models along the contact surfaces.

When the hexahedral element is used for shell elements, the ζ direction must be the thickness direction. The dimension along this direction is much smaller than other dimensions.

For the interpolation of coordinates in pentahedral elements, we can use linear pentahedral element (Figure 3.13), quadratic pentahedral (Figure 3.14), or cubic Pentahedral interpolation.

When the pentahedral element is used for shell elements, the ζ direction must be the thickness direction. The dimension along this direction is much smaller than other dimensions.

For the interpolation of coordinates in tetrahedral elements, we can use linear tetrahedral element (Figure 3.15), quadratic tetrahedral (Figure 3.16), or cubic tetrahedral interpolation.

Several options are available for the displacement interpolation in a three-dimensional element. The Lagrangian, Pentahedral and Tetrahedral elements are conventional elements. The FQP elements combine a 2D element in cross-section with an axode at every 2D node location. Each axode may have a different order (or number of nodes). These FQP elements have only

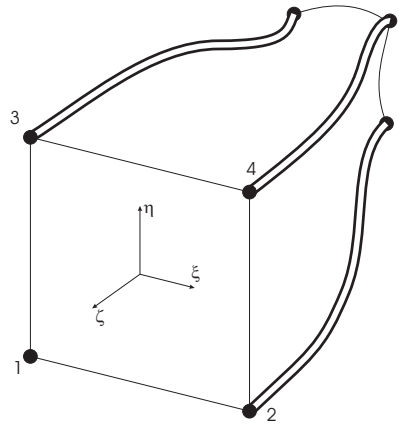

Figure 3.9: Linear FQP displacement element

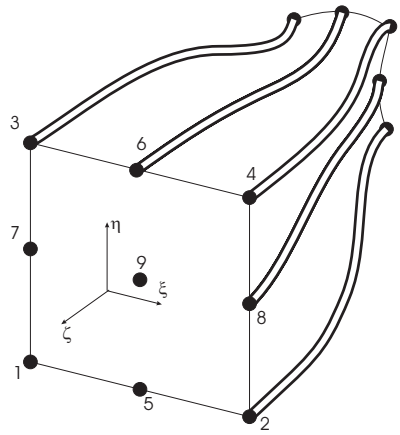

Figure 3.10: Quadratic FQP displacement element

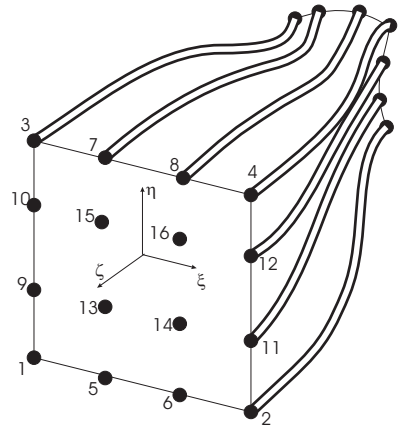

Figure 3.11: Cubic FQP displacement element

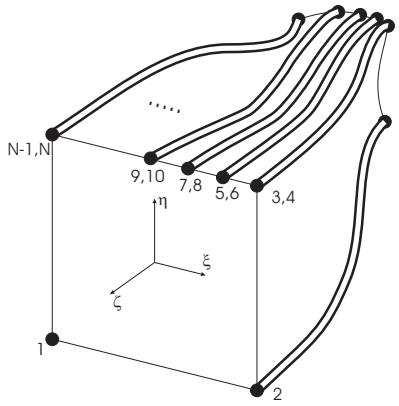

Figure 3.12: Surface FQP coordinate element

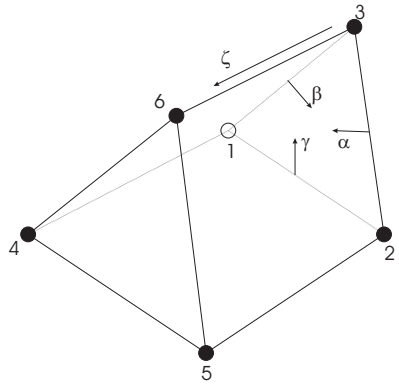

Figure 3.13: Linear pentahedral coordinate element

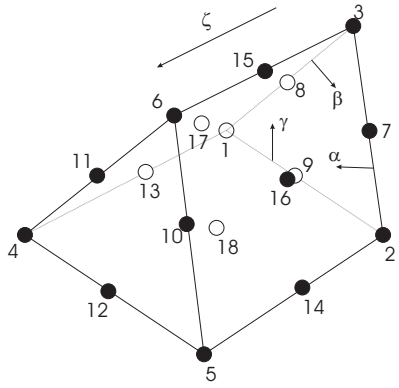

Figure 3.14: Quadratic pentahedral coordinate element

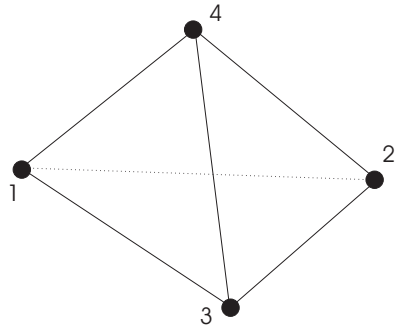

Figure 3.15: Linear tetrahedral coordinate element

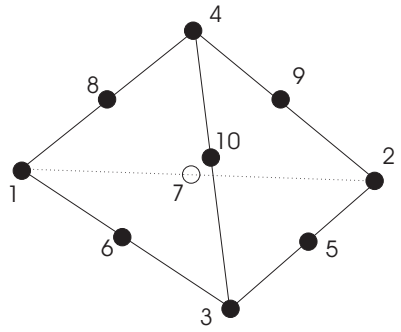

Figure 3.16: Quadratic tetrahedral coordinate element

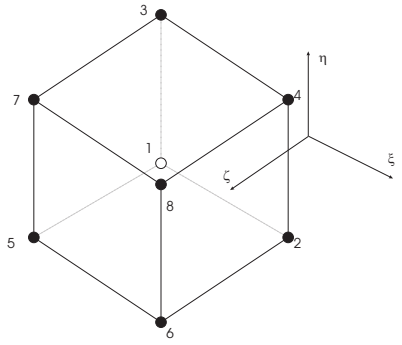

Figure 3.17: Linear Lagrangian hexahedral displacement element

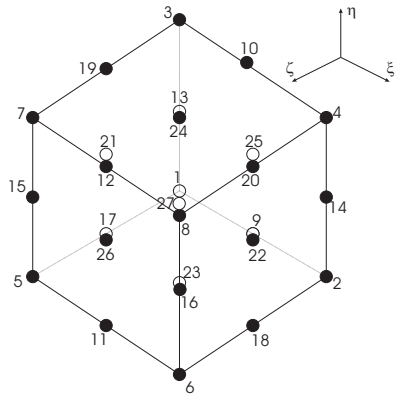

Figure 3.18: Quadratic Lagrangian hexahedral displacement element

been implemented for the rectangular cross sections.

For the interpolation of displacements in solid hexahedral elements, we can use linear Lagrangian (Figure 3.17), quadratic Lagrangian (Figure 3.18), cubic Lagrangian, Linear FQP (Figure 3.19), Quadratic FQP (Figure 3.20), and cubic FQP (Figure 3.21) interpolation.

For the interpolation of displacements in solid pentahedral elements, we can use linear pentahedral (Figure 3.22), quadratic pentahedral (Figure 3.23), or cubic pentahedral interpolation.

Similarly, for the interpolation of displacements in solid tetrahedral elements, we can use linear tetrahedral (Figure 3.24), quadratic tetrahedral (Figure 3.25), or cubic tetrahedral interpolation.

Shell elements use 2D interpolation in cross section for displacement interpolation. At every 2D node location, there is an option of using an axode with variable number of nodes, or of using 2 nodes at the middle cross-section. If 2 nodes are used, the first of this pair is associated with the 3 translation degrees of freedom at the mid-section and the second node is associated with the 3 rotation degrees of freedom at the mid-section.

When the hexahedral element is used with shell interpolation, the interpolation schemes based on a the 2D quadrilateral element is used. Either linear (Figure 3.26), quadratic (Figure 3.27) or cubic shell elements (Figure 3.28) can be formed.

Similarly, when a pentahedral element is used with shell interpolation, a 2D triangular interpolation is used in cross-section. The triangle may be linear (Figure 3.29), quadratic (Figure 3.30) or cubic (Figure 3.31).

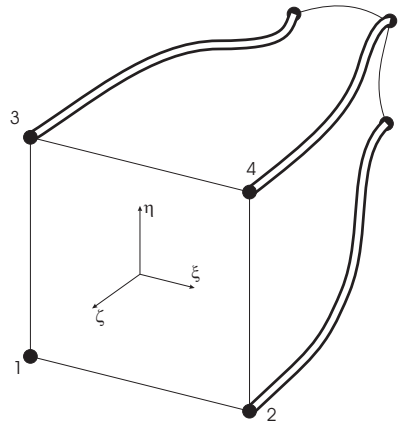

Figure 3.19: Linear FQP displacement element

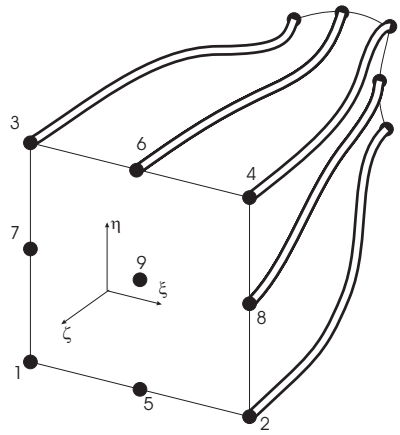

Figure 3.20: Quadratic FQP displacement element

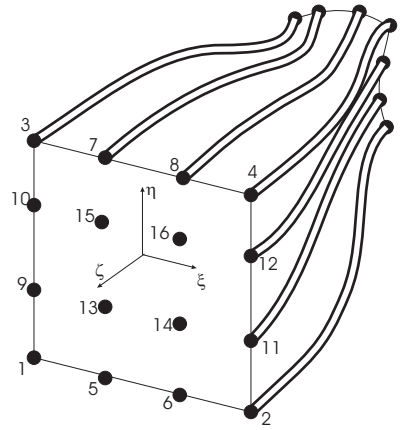

Figure 3.21: Cubic FQP displacement element

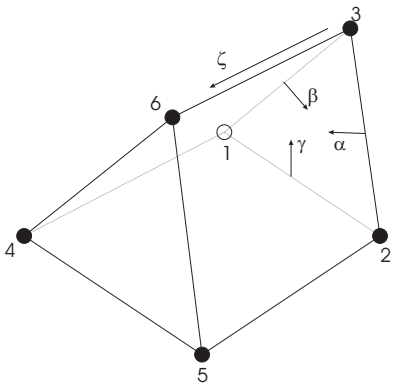

Figure 3.22: Linear pentahedral displacement element

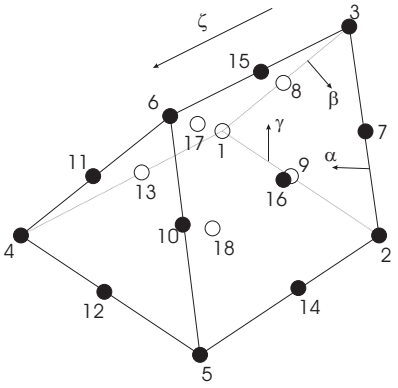

Figure 3.23: Quadratic pentahedral displacement element

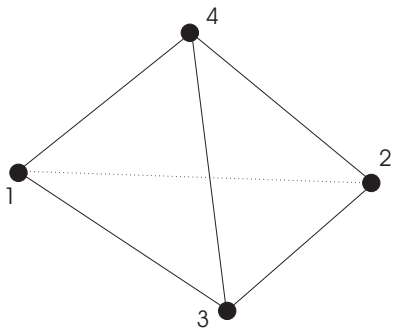

Figure 3.24: Linear tetrahedral displacement element

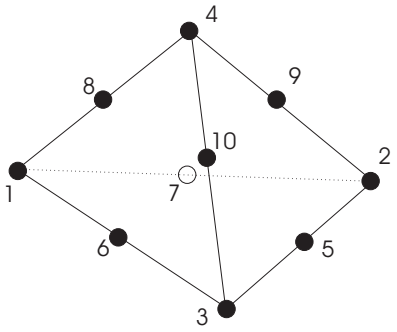

Figure 3.25: Quadratic tetrahedral displacement element

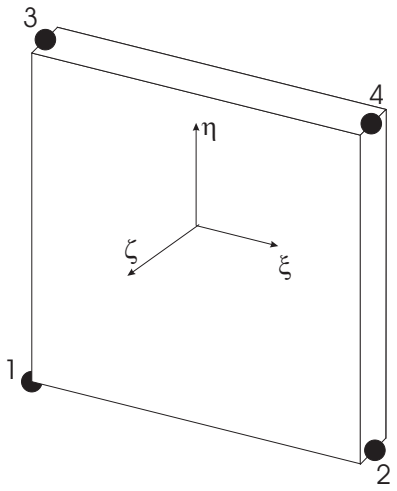

Figure 3.26: Linear shell displacement element. The displacement field at each numbered vertex can be represented either by a displacement axode, or by a pair of displacement nodes.

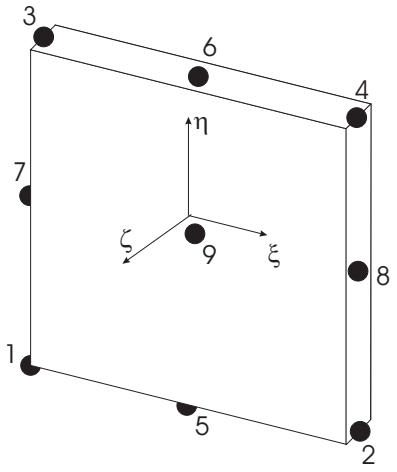

Figure 3.27: Quadratic shell displacement element. The displacement field at each numbered vertex can be represented either by a displacement axode, or by a pair of displacement nodes.

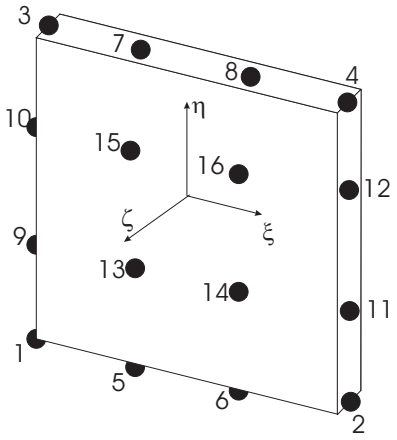

Figure 3.28: Cubic shell displacement element. The displacement field at each numbered vertex can be represented either by a displacement axode, or by a pair of displacement nodes.

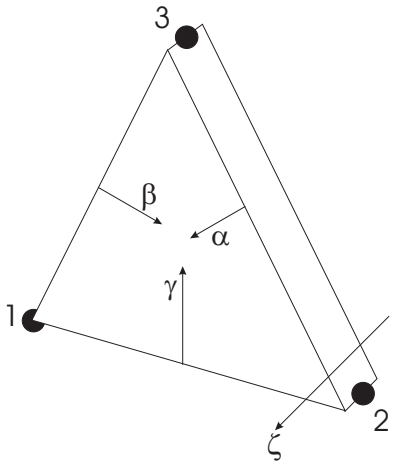

Figure 3.29: Linear pentahedral shell displacement element. The displacement field at each numbered vertex can be represented either by a displacement axode, or by a pair of displacement nodes.

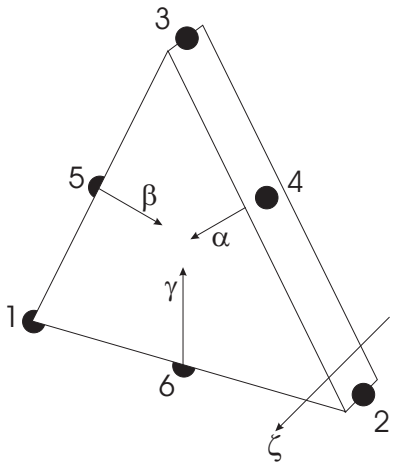

Figure 3.30: Quadratic pentahedral shell displacement element. The displacement field at each numbered vertex can be represented either by a displacement axode, or by a pair of displacement nodes.

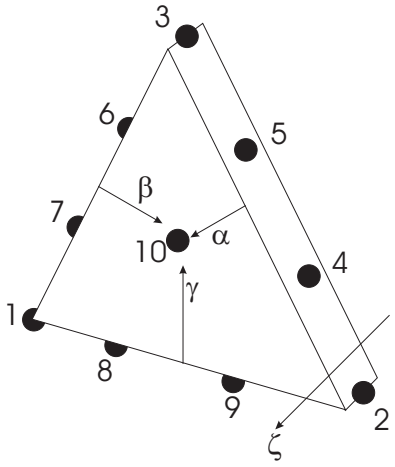

Figure 3.31: Cubic pentahedral shell displacement element. The displacement field at each numbered vertex can be represented either by a displacement axode, or by a pair of displacement nodes.

3.1.5 Degree of Freedom Sets

The degrees of freedom of the finite element models are organized into sets, which we refer to as 'Dofsets' (short for Degree of freedom sets). Each Dofset can be assigned a name. Degrees of freedom that are not explicitly assigned a Dofset are collected into a default Dofset named 'Slave'. *Calyx* provides mechanisms for applying constraints to Dofsets by name. Dofsets may also be picked out by name for static condensation.

Optionally, an entire Dofset can be constrained to move along a cylindrical, planar or conical surface whose deformation is expressed in terms of a Fourier and a polynomial series. This 'Fourier' Dofset technique is particularly useful in joining two incompatible finite element meshes at a common interface.

Special Dofsets called rigid regions can be used to describe the rigid body type motions of rigid parts of a model. A mechanism has been provided to connect nodal degrees of freedom of adjacent finite elements to this rigid region. This technique is used to connect a gear model to a rigid region such as a rigid bearing race.

3.1.6 Structures

The deformation behavior of the system is represented by a *Structures*. Each structure can be used or instantiated many times. Properties common to all instances of a structure are stored in one place, called the *prototype* of the structure.

Structure prototypes are built recursively using a hierarchy of other prototypes. Individual finite element meshes and external structures form the lowest level in this hierarchy. The slave degrees of freedom of all lower level structures are statically condensed out before being assembled into a higher level prototype. This hierarchical structural assembly and condensation is what makes it possible to analyze the extremely large system models described later in this report.

3.1.7 Bodies

The gear system is divided into a number of moving entities that we call 'bodies' (Figure 3.32). Each body can undergo large rigid body type translations and rotations. The bodies interact with each other through contact and 'bearing' connections.

Each body is assigned a name, a structure prototype, and a reference frame.

The structure prototype encapsulates the deformation behavior of the body. It is only when a prototype structure is assigned to a body in this manner that copies of that structure are actually created. This is called 'instantiating' the structure. A particular prototype may be instantiated any number of times in different bodies, or might not be instantiated at all. Upon instantiation a separate copy of the prototype, called an instance, is created. For each individual instance of a high level prototype, a new instance of all its next lower level prototypes is created. Thus each body has its own unique structure instance. It also has a structure prototype, but this prototype is not necessarily unique.

Kinematic information is associated with the body's reference frame by providing a symbolic expression for the kinematic transformation from the ground reference frame to the body's reference frame. This transformation is time-dependent for moving bodies.

The actual location of the reference frame assigned to a body depends not only on the kinematics of the system, but also on the deformation of the system. The deviation of the location of the reference frame from its kinematic position is called the reference frame displacement (Figure 3.33). The reference frame has six degrees of freedom and therefore has six displacement components. Each component can either be constrained to a prescribed value or allowed to freely attain its equilibrium displacement. The reference frame shown in Figure 3.34 has two constraints. Its U_x and θ_z degrees of freedom are constrained to zero.

The free degrees of freedom can be loaded with external forces of moments (Figure 3.35).

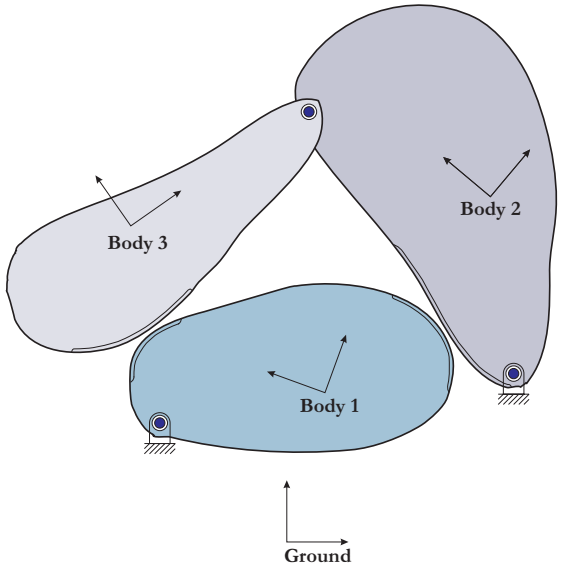

Figure 3.32: Multi-body system representation inside of *Calyx*

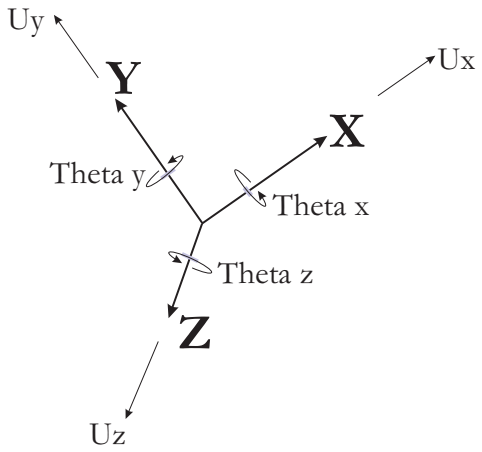

Figure 3.33: Reference frame displacements

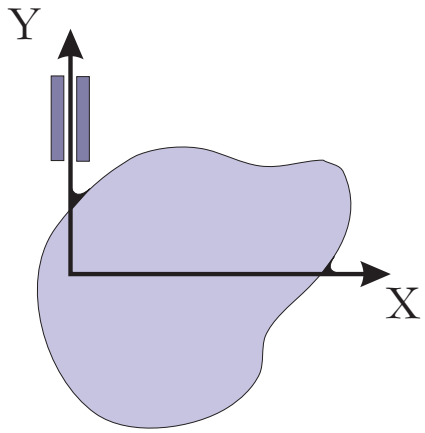

Figure 3.34: Reference frame constraints

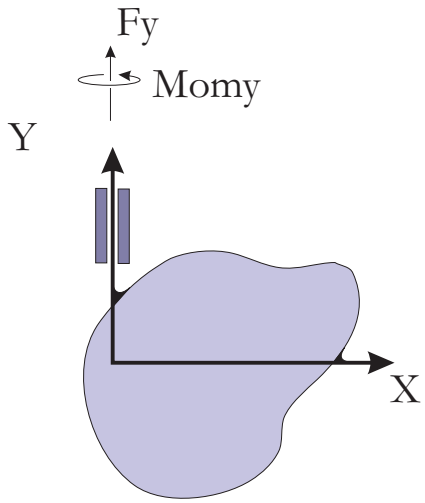

Figure 3.35: External loads applied to unconstrained reference frame degrees of freedom.

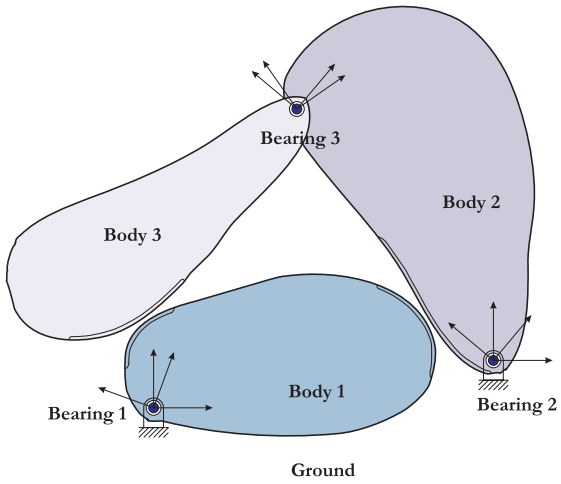

Figure 3.36: Bearing connections in the multi-body model.

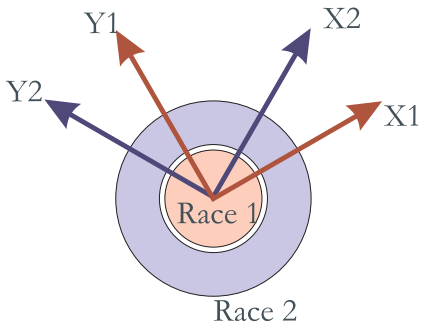

Figure 3.37: Bearing races

For the constrained reference frame degrees of freedom, reaction forces will be developed which will force the reference frame to attain the prescribed displacements. For the free reference frame degrees of freedom, the reaction forces will, of course, be equal to the prescribed loads. These reactions may be accessed after solution.

3.1.8 Bearing Constraints

Other than the contacting surfaces, bearings are the only means for the various bodies in the gear system to interact with each other (Figure 3.36).

One way of modeling a bearing is to treat the outer and inner races and all the rolling elements as independent bodies interacting through their contacting surfaces.

A simpler method may be used when stiffness values are available for the bearings. Each race (Figure 3.37) is treated as a rigid region, and the six degrees of freedom of the first bearing race and the six degrees of freedom of the second bearing race are inter-related through a stiffness matrix.

Like bodies, prototypes, and instances, bearings are also assigned reference frames. Each bearing has two reference frames, one for each bearing race. Each race can be attached either to the ground, or to one of the moving bodies.

When a race is attached to the ground, only a transformation operator from the ground reference frame to the race reference frame is needed to define the race.

If a bearing race is attached to a rigid region of a body, then in order to set up the race, we need the body, the name of the rigid region Dofset and a transformation operator from the body reference frame to the bearing race reference frame.

The bearing race transformation operator cannot have any time dependencies.

The axis of rotation of the bearing, by convention, will be the Z axis of the bearing race reference frames. The origins and the directions of the Z axes of the two race reference frames must coincide.

A stiffness matrix $\mathbf{K}$ and a damping matrix $\mathbf{C}$ is assigned to each bearing when it is created.

Normally the bearing behavior is of the type:

$$\mathbf{K}\mathbf{x} + \mathbf{C}\dot{\mathbf{x}} = \mathbf{f} \tag{3.1}$$

In some circumstances, we would like the bearing to assume a deformed shape $\mathbf{x} = \mathbf{x}_o$ under no load. It would obey a relationship of the type:

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_o) + \mathbf{C}\dot{\mathbf{x}} = \mathbf{f} \tag{3.2}$$

This type of behavior can be imposed on the bearing through the 'unloaded deformation' feature we have implemented..

It might sometimes be necessary to specify an additional *external* load $\mathbf{f}_{ext}$ acting on the bearing:

$$\mathbf{K}\mathbf{x} + \mathbf{C}\dot{\mathbf{x}} = \mathbf{f} + \mathbf{f}_{ext} \tag{3.3}$$

This external load is introduced using the 'undeformed load' feature.

The undeformed loads and unloaded deformations are a convenient way to introduce pre-load in bearings, to misalign a bearing, or to introduce torque into a model of a recirculating power system.

Sometimes, we need to impose constraints of the type:

$$\delta + a_{U_x}U_x + a_{U_y}U_y + a_{U_z}U_z + a_{\theta_x}\theta_x + a_{\theta_y}\theta_y + a_{\theta_z}\theta_z \geq 0 \tag{3.4}$$

on bearing degrees of freedom. This would happen, for instance, in a bearing that has radial or axial clearance. We allow this through a 'bearing gap constraint' feature.

3.1.9 Contact Surfaces and Surface Pairs

Pairs of surfaces in the system can make contact with each other, as shown in Figure 3.38. In order to analyze this contact, a contact grid must be set up for each surface pair.

Figure 3.39 shows a computational grid that has been set up in the contact zone of a gear tooth. The entire face width of the tooth is divided into $2N + 1$ slices. N is a user selectable quantity. For each slice $j = -N : +N$, a cross section of the surface is taken at the middle of the slice, and a point is located on this slice that approaches the surface of the mating surface the closest. This selection is carried out using the undeformed geometry. If the separation between the two surfaces at this closest point is larger than a user selectable separation tolerance ϵ, then the entire slice is eliminated from further consideration. Otherwise, a set of grid cells identified by the grid cell location indices $(i, j), i = -M : M$ is set up centered around this closest point of slice j. The number M is user selectable. The dimension of the grid cells in the profile direction Δs is also user selectable. Here s is the curve length parameter measured along the surface 'profile'. In three-dimensional *Calyx* models, s is a dimensional quantity in length units. However, for historical reasons, in two-dimensional models, s is a dimensionless curve length parameter varying between 0 and $N_{pts} - 1$ where N_{pts} is the number of coordinate nodes used over the length of the surface 'profile'.

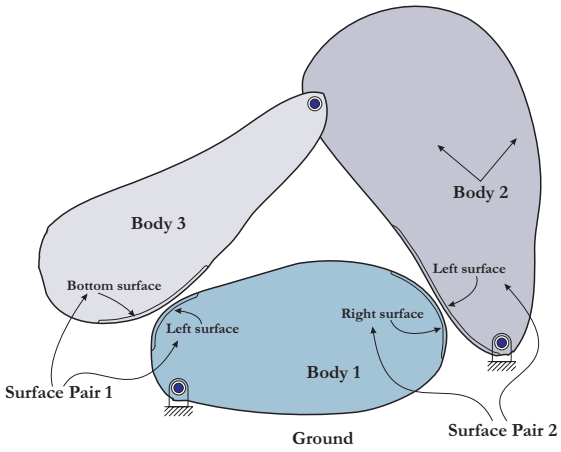

Figure 3.38: Contact surfaces and surface pairs in the multi-body model

The width of the grid is $(2M + 1)\Delta s$. Choosing the correct width is crucial for obtaining correct contact pressures. Using too wide a grid for a fixed M can result in loss of resolution, because only the center grid cell will end up carrying all the load (Figure 3.40). If the grid is too narrow, then the contact zone will get truncated, causing artificially high contact pressures at the edges of the grid (Figure 3.41). If the grid size is correct, a variation of contact pressure similar to that in Figure 3.42 should be obtained.

The contact surface pairs and their grid parameters are specified by the user before the analysis is started.

3.1.10 Analysis

The analysis can be carried out in static, quasi-static, or dynamic mode. For three-dimensional models we prefer not to use the dynamic mode. For two-dimensional models, we use all three methods.

Before analysis, the stiffness and mass matrices for the whole system are built and decomposed once. These do not change from one time step to another.

For each time step, contact grids are set up on all the contact surface pairs. A unit contact load is applied at one grid cell, and a backsubstitution is carried out for the whole system to compute its response to the unit load. The increase in separation at all grid cells due to this unit load is obtained from the system response. The response is saved. Then a unit load is applied to the next grid cell, and the process is repeated. After the backsubstitution is done for all grid cells, a system compliance matrix is populated with the results.

The contact equations are then formed from the compliance matrix and sent to a contact solver. The contact solver solves the equations using a method related to the Simplex method of linear programming. In most situations, the unique solution to the contact problem will be found in a fixed number of steps.

Under certain circumstances, however, the contact algorithm will flag an error and terminate. One such error condition will occur if any of the bodies in the system has a reference frame that is insufficiently constrained or if the body does not possess adequate contact constraints. Then the contact solver will fail in its attempt to equilibrate the multibody system.

The second error condition will occur if multiple solutions exist to the contact equations. This can occur if any set of the gap constraint equations is not independent. This condition will cause a loss of rank in the gap compliance matrix, and is easily detected.

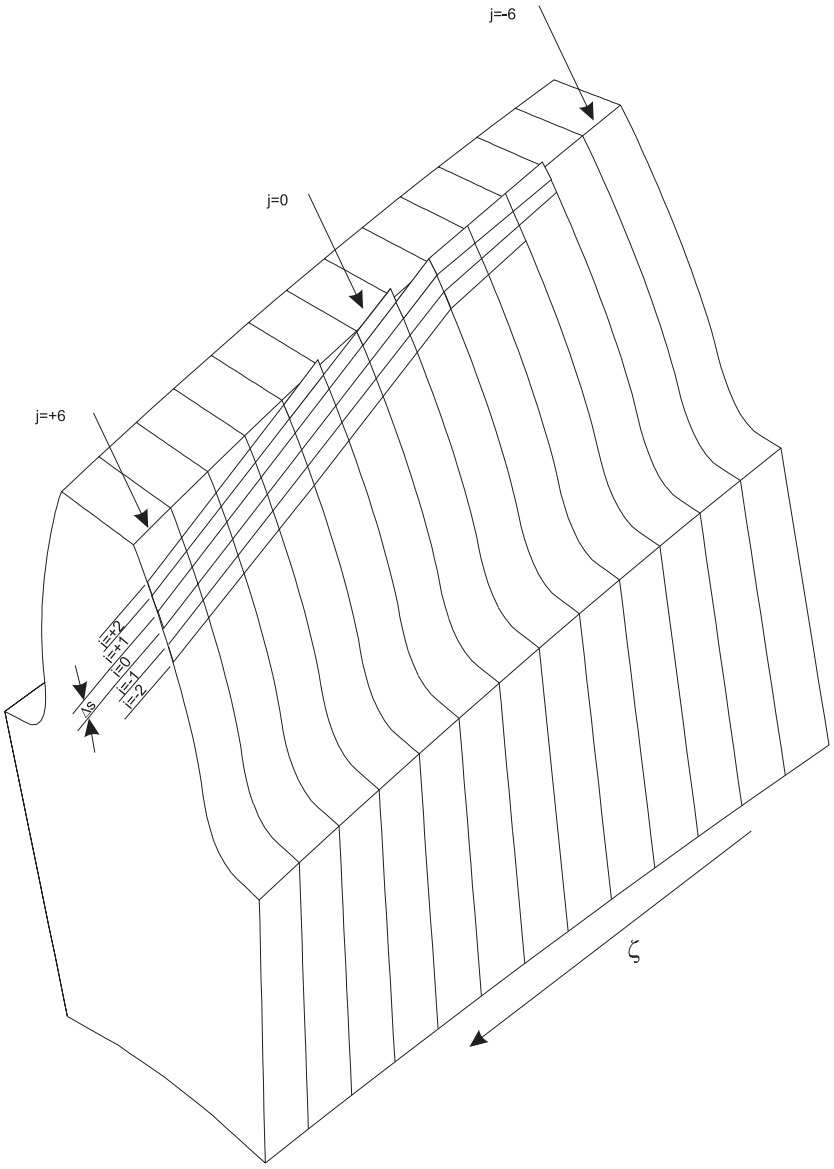

Figure 3.39: Computational grid in the contact zone of the gears

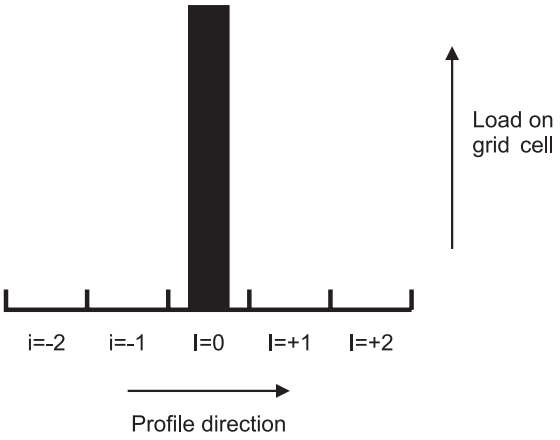

Figure 3.40: Contact pressure distribution across the width of contact obtained when the contact grid is too wide.

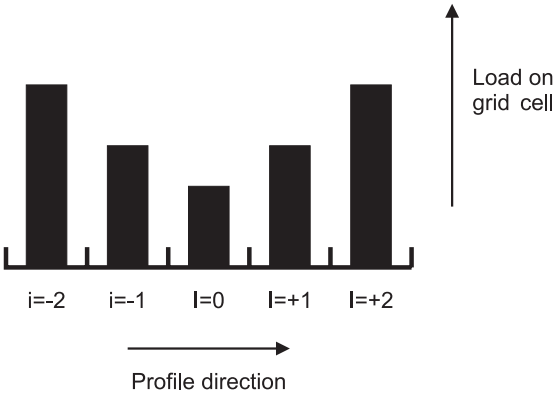

Figure 3.41: Contact pressure distribution across the width of contact obtained when the contact grid is too narrow.

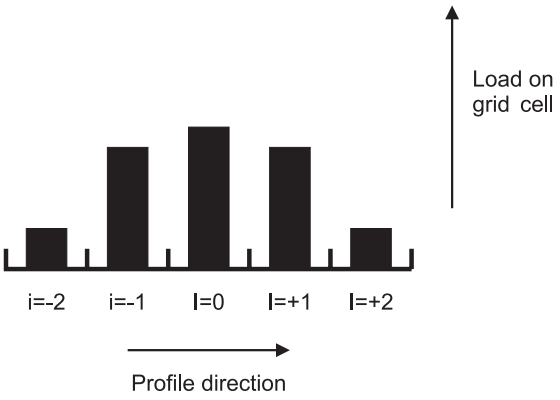

Figure 3.42: Contact pressure distribution across the width of contact obtained when the contact grid is correct.

3.1.11 Postprocessing

The results we are usually interested in obtaining from an analysis are the stress values at fillets of gear teeth, contact pressures values, pictures of load distribution, deformed shapes, body reaction forces and moments, body deflections, bearing deformations, and bearing reactions.

Some of these results are written out to ASCII files after analysis is completed for each time step. These include body deflections, body reactions, bearing deflections, an bearing reactions.

We have provided a means to instrument the model with load, stress and displacement sensors. The readings of these sensors are also written out immediately after analyzing each time step.

There is also a mechanism for *Calyx* to dump the entire state of the model, including all displacement, stress, and contact information to a large post-processing data file after each time step. This post-processing file can be opened during a post-processing stage, and a variety of plots can be generated. Most of the 3D views, the stress contours, X-Y plots of contact pressure and stress vs. time, and the transmission error plots shown in the following sections of this report were generated this way.

3.2 Multyx

Multyx serves as a human-friendly interface between *Calyx* and the user. It provides the user with a multi-level menu hierarchy that contains commands to set analysis parameters, model data, and post-processing options.

3.2.1 System Definition Files

For each software package such as *Planetary2D*, *Helical3D*, and *HypoidFaceMilled*, *Multyx* needs to know what kind of data is required from the user, how to build the menu system for that package, and how to build the model once it has the data. All this information is kept in a special data file called a *System Definition File* or SDF file.

The following is a section of the SDF file of the *HypoidFaceMilled* package:

```
%FLOAT%
%NAME%H3CONCAVEPINION
%INITIAL%0.0
%DEFAULT%0.0
%DESCRIPTION%Helical motion 3rd order coefficient.
%HELP%
Helical Motion 3rd Order Coefficient: The
sliding base position of the workpiece Xb is
related to the cradle motion Phi through the
following relation when helical motion is
included:
Xb=Xbo+H1*Phi + (1/2)H2*Phi^2 +(1/3)H3*Phi^3
H3 is the 3rd order Helical Motion Coeff.
Special Analysis Record #:16  Item#:5
%ENDHELP%
%ENDFLOAT%
```

It contains the description of the data item H3CONCAVEPINION. This description tells *Multyx* that this is a floating point item. It is to have an initial value of 0.0. It has a default value of 0.0. There is a description of the variable, and some lines of help information that can be displayed to the user if necessary.

Multyx supports many different types of data items, including floating point items, integers, character strings, file names, enumerated types, and boolean values. It also supports multidimensional arrays of these basic types.

After the list of data items needed to define the system, the SDF file has a description of all the menus that the user must use to type in the data. The following is another section of the SDF file from the *HypoidFaceMilled* package:

```
%MENU%
%NAME%MACHINECONVEXPINION
%LABEL%MACHINE
%PROMPT%Machine
%PARENT%CONVEXPINION
%DESCRIPTION%Machine settings.
%ITEMS%
RADIALCONVEXPINION
RADIALSETTING
TILTCONVEXPINION
TILTANGLE
SWIVELCONVEXPINION
SWIVELANGLE
BLANKOFFSETCONVEXPINION
BLANKOFFSET
ROOTANGLECONVEXPINION
ROOTANGLE
MACHCTRBACKCONVEXPINION
MACHCTRBACK
SLIDINGBASECONVEXPINION
SLIDINGBASE
CRADLECONVEXPINION
CRADLEANGLE
RATIOROLLCONVEXPINION
RATIOROLL
C2ROLLCONVEXPINION
2C
C3ROLLCONVEXPINION
6D
C4ROLLCONVEXPINION
24E
C5ROLLCONVEXPINION
120F
H1CONVEXPINION
H1
H2CONVEXPINION
H2
H3CONVEXPINION
H3
V1CONVEXPINION
V1
V2CONVEXPINION
V2
V3CONVEXPINION
V3
```

```
%ENDITEMS%
%ENDMENU%
```

This section contains the name (MACHINECONVEXPINION) of the menu, and the name (CONVEXPINION) of its parent menu. MACHINE is the short label that is displayed to the user. By assigning a parent to each menu, *Multyx* can build the complete menu hierarchy. Between the lines %ITEMS% and %ENDITEMS% is a list of data items, including the item H3CONCAVEPINION. For each item is a shorter label (like H3) which is displayed to the user.

After the user is done traversing all the menus entering the data for a system, *Multyx* needs to generate the finite element model, bearing files, and other components of the *Calyx* model. The user initiates this step by issuing a GENERATE command to *Multyx*. *Multyx* then reads a part of the SDF file called a 'Generator Script' and interprets it. This generator script is in the form of a computer program. The following is a section of the generator script for the *HypoidFaceMilled* package:

```
var CutterToWorkpieceXfm;
var s1,s2,t1,t2;
if(isLeftHanded) {
  CutterToWorkpieceXfm:=[ConcaveCutterToWorkpieceXfm,ConvexCutterToWorkpieceXfm ];
  s1=0.5*EDGERADIUSCUTTERCONCAVEPINION*pi;
  s2=0.5*EDGERADIUSCUTTERCONVEXPINION*pi;
  s1=((rpt-rrt)^2+(zpt-zrt)^2)^0.5;(FACEANGLEPINION-PITCHANGLEPINION)*Degree*(OUTERCONEDISTPINION-
  s2=((rpt-rrt)^2+(zpt-zrt)^2)^0.5;(FACEANGLEPINION-PITCHANGLEPINION)*Degree*(OUTERCONEDISTPINION-
  t1=(-SWIVELCONCAVEPINION+CRADLECONCAVEPINION-SPIRALANGLEPINION)*Degree;
  t2=(-SWIVELCONVEXPINION +CRADLECONVEXPINION -SPIRALANGLEPINION)*Degree;
} else {
  CutterToWorkpieceXfm:=[ConvexCutterToWorkpieceXfm ,ConcaveCutterToWorkpieceXfm];
  s1=0.5*EDGERADIUSCUTTERCONVEXPINION*pi;
  s2=0.5*EDGERADIUSCUTTERCONCAVEPINION*pi;
  s1=((rpt-rrt)^2+(zpt-zrt)^2)^0.5;(FACEANGLEPINION-PITCHANGLEPINION)*Degree*(OUTERCONEDISTPINION-
  s2=((rpt-rrt)^2+(zpt-zrt)^2)^0.5;(FACEANGLEPINION-PITCHANGLEPINION)*Degree*(OUTERCONEDISTPINION-
  t1=( SWIVELCONVEXPINION -CRADLECONVEXPINION +SPIRALANGLEPINION)*Degree;
  t2=( SWIVELCONCAVEPINION-CRADLECONCAVEPINION+SPIRALANGLEPINION)*Degree;
}
var nFaceDivs;
if(MESHTYPE==1) {
  nFaceDivs=1;
} else {
  nFaceDivs=NFACEDIVSPINION;
}
Inform("Generating pinion tooth finite element mesh...");
GenerateToothMeshFromSurfaceEnvelope(
  NTEETHPINION,
  meshfilename,tplfilename,DoubleSided,
  CutterDefn,phi,CutterToWorkpieceXfm,circthickness,
  OuterCone,BaseCone,BackCone,FrontCone,PitchCone,ThicknessCone,
  MESHTYPE,LIMDSPORDPINION,LIMITCOORDORDERPINION,nFaceDivs,
  // Whether the Envelope parameter & two cutter surface parameters
  // are angular values:
  TRUE,FALSE,TRUE,
  //Starting guesses for Envelope parameter & two cutter surface parameters for Side1:
  0,s1,t1,
  //Starting guesses for Envelope parameter & two cutter surface parameters for Side2:
```

```
    0,s2,t2,
    // Material Properties:
    YOUNGSMODPINION,POISSONPINION,DENSITYPINION,RALEIGHALPHAPINION,RALEIGHBETAPINION,
    SPECIFYMSRPTPINION,rthick,zthick
  );
  Inform("Done generating pinion tooth finite element mesh.");
```

It shows some of the steps involved in choosing the cutter definition, machine kinematics, and generating the finite element model of the pinion. This script written in the *Calyx* programming language described earlier. *Multyx* has the ability to interpret such a script.

After *Multyx* executes this script, the system model is ready for *Calyx* to analyze. The user initiates the analysis step by invoking the 'STARTANAL' command, and *Multyx* starts a copy of *Calyx* running in the background, and gives it the command necessary to start analysis. When it is done, the *Multyx* gives control back to the user.

Postprocessing is done through a set of hard-wired postprocessing menus that user accesses through the POSTPROC command. The user can use the commands and sub-menus under POSTPROC to obtain all the graphs, drawings, reports and 3D views.

3.2.2 Template Files

In order to build the gear tooth models, *Multyx* needs the number of finite elements, and the nodal connectivity information for each finite element. Instead of having the user provide this information manually, we have pre-packaged this information into several 'template' files. Each template file has the connectivity information for a different level of mesh refinement. The user simply selects one of these template files for each gear tooth. Figures 3.43 through 3.46 show the four standard templates.

3.3 Guide

Guide provides the modern graphical user interface (Figure 3.47) to *Multyx*. The streams of informational, error and warning messages that *Multyx* sends to the user are separated by *Guide*, and presented in separate windows as shown on the upper right hand side of Figure 3.47. The user can activate these message windows by clicking the appropriate "Error", "Information" or "Warning" tab. Graphical information sent by *Multyx* is directed to a graphics window shown on the lower right hand side.

In Figure 3.47, the large buttons such as those labeled EXIT ,QUIT, OPTIONS, LOADSESSION, EDIT send commands to *Multyx* when hit by the user. In response to the command, *Multyx* might carry out an action, as in the case of the LOADSESSION command, or lead the user to a different menu, as in the case of the EDIT command. Moving the mouse over a button without depressing it will cause *Guide* to momentarily pop up a balloon (a tool tip) containing a short description of the use of that button.

Integer data items are entered through a dialog box of the kind shown in Figure 3.48. The current value appears in a box in the dialog box. If the value of the data item is undefined, then the box appears blank.

Floating point data is entered through the dialog box shown in Figure 3.49.

Boolean data items are those that can only take a YES/NO or TRUE/FALSE type of value. Their value is set by checking or clearing the box as shown in Figure 3.50.

String data items contain ASCII strings. The dialog box shown in Figure 3.51 allows the user to enter string type data.

The last kind of data item is of the 'switch' type. The value of a switch menu item can be selected from a fixed set of valid choices. The choice is made through a drop down list as shown in Figure 3.52.

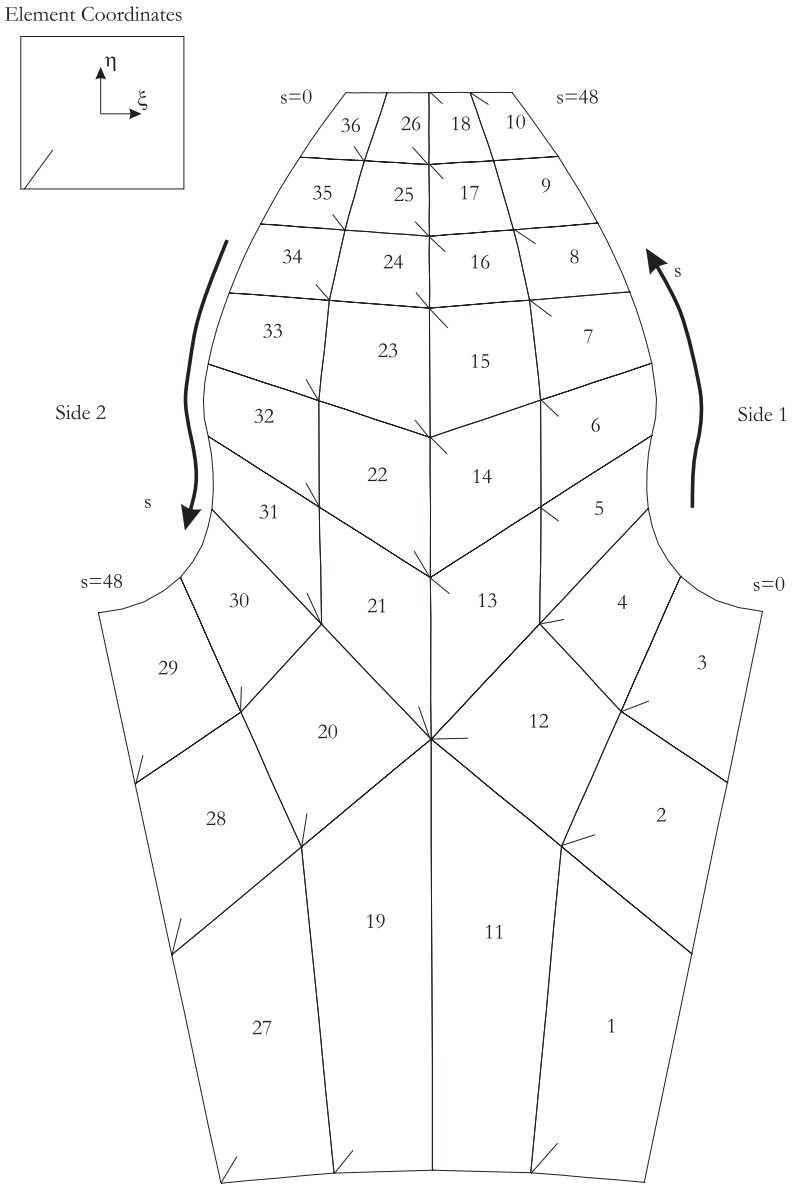

Figure 3.43: The MEDIUM.TPL template.

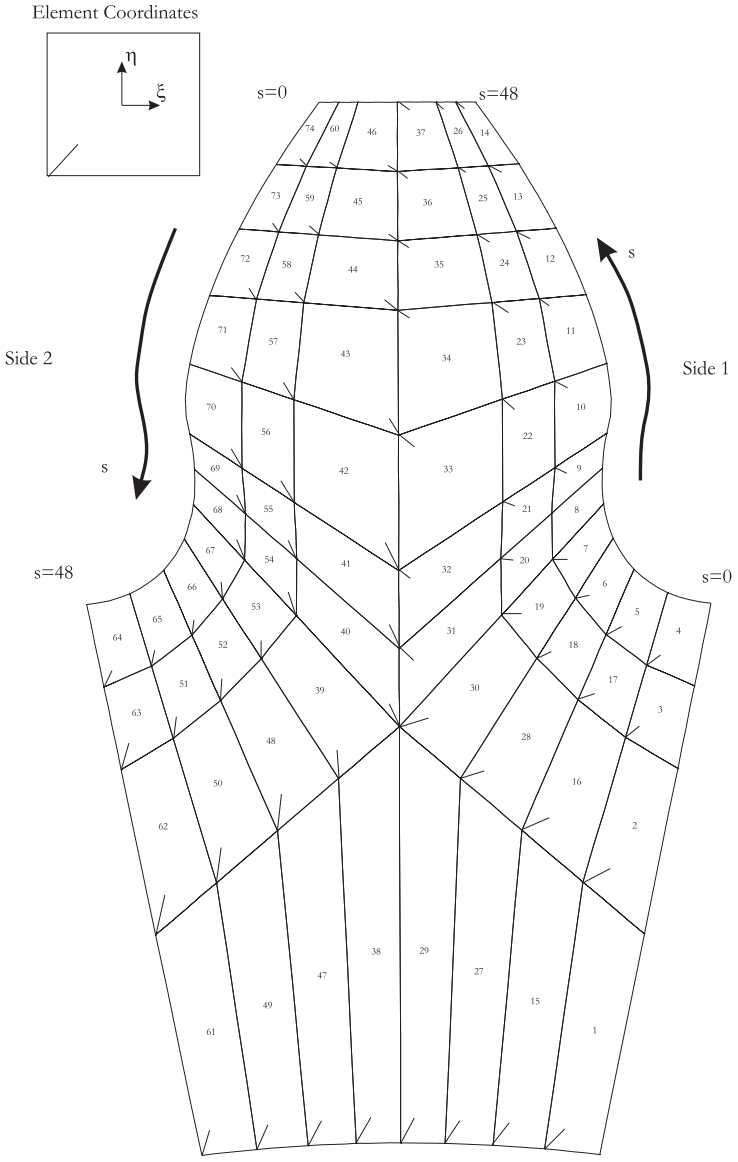

Figure 3.44: The FINEROOT.TPL template.

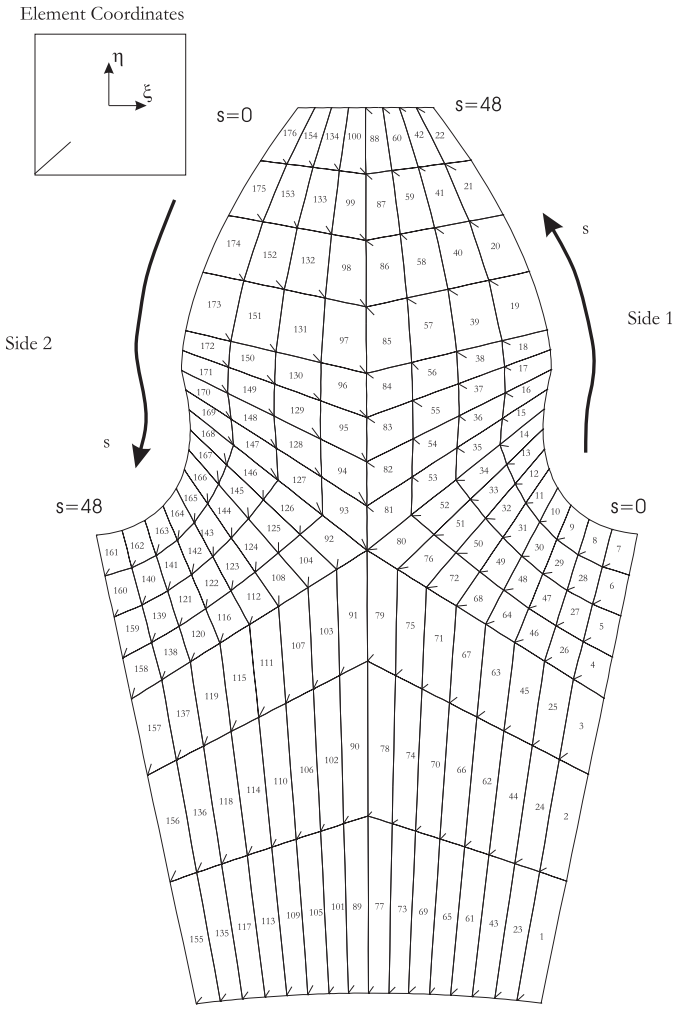

Figure 3.45: The FINEST.TPL template.

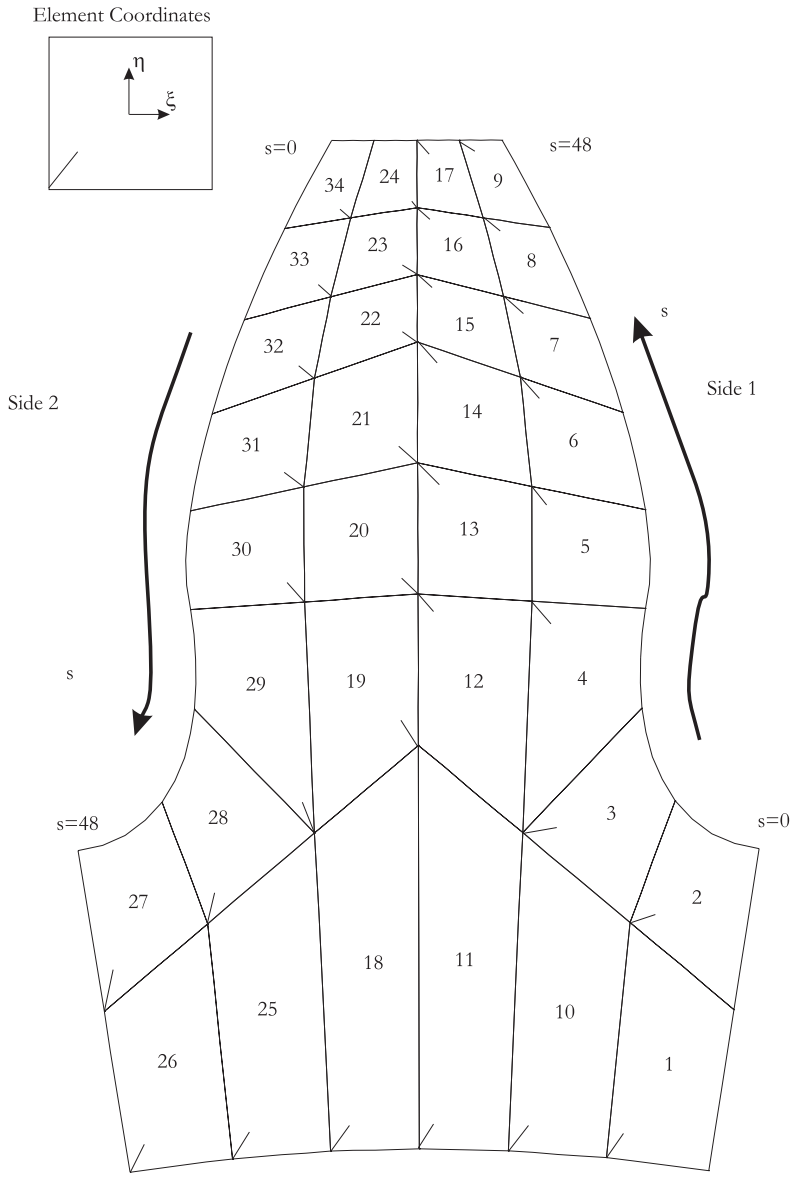

Figure 3.46: The THINRIM.TPL template.

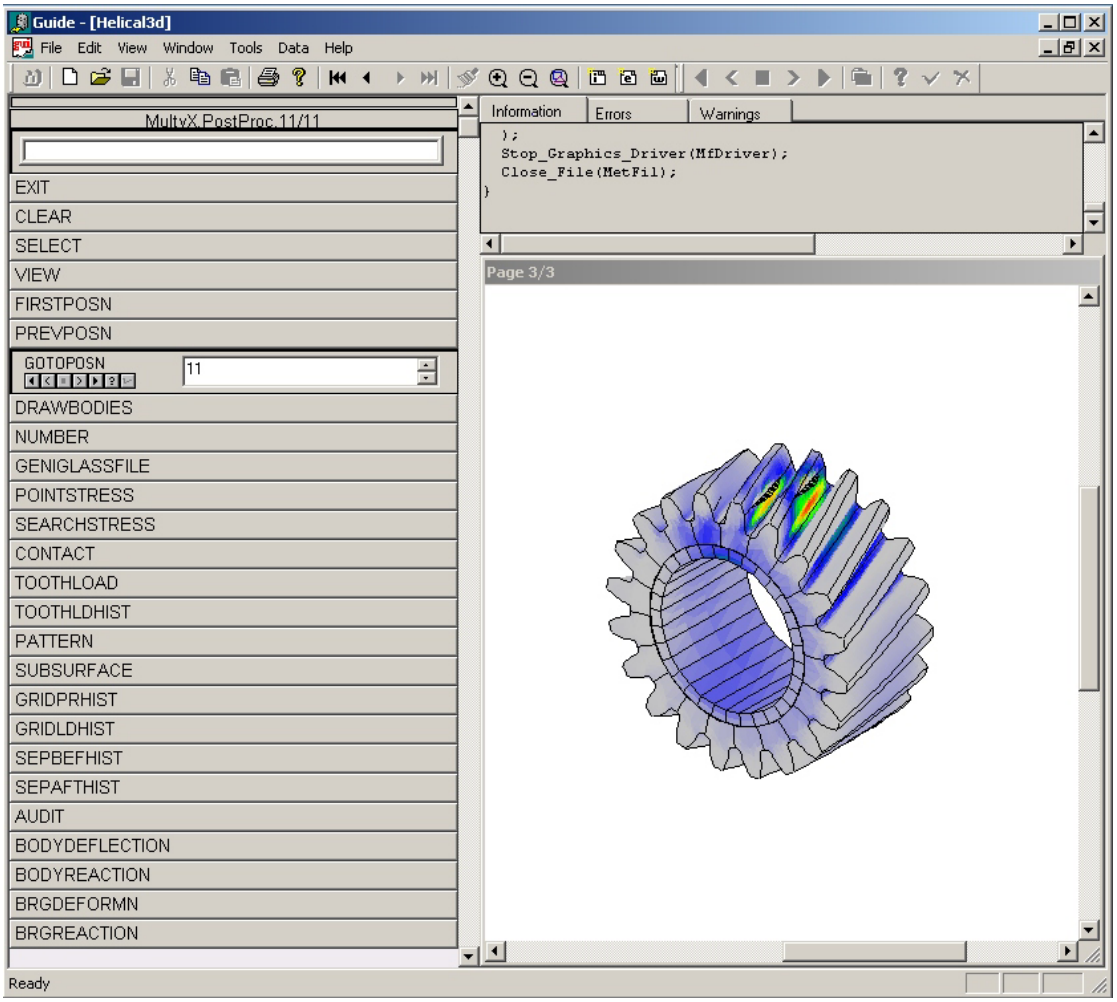

Figure 3.47: The user interface provided by *Guide*.

Figure 3.48: An integer data entry box

Figure 3.49: An floating point data entry box

Figure 3.50: An boolean data entry box

Figure 3.51: An string data entry box

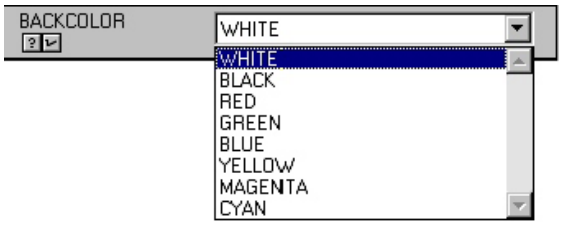

Figure 3.52: An switch type data entry box

The data entry dialog boxes use a few small buttons as short cuts for common tasks as shown in the Table 3.3. Some of these buttons may be disabled depending upon the particular item and its value.

Guide directs the graphical output from *Multyx* to a graphics window. This graphics can be copied and pasted into other programs, or saved as Encapsulated PostScript (EPS) files.

Guide is an important component in providing the users with interface features they expect from a modern computer program.

3.4 iGlass

iGlass is a powerful tool for pre- and post-processing gear models and results. It uses hardware graphics acceleration in the computer to display highly interactive three-dimensional views of the model. The model can interrogate this view in many ways, and can interactively rotate, zoom and pan on the model. Figure 3.53 shows a gear through *iGlass*.

The finite element mesh(Figure 3.54) of the model can be displayed by checking the 'Finite Element Mesh' box.

A cutting plane feature is available that allows the user to cut the model by a cutting plane. This feature is useful in pre and post processing of complicated models with a large number of internal gears. The model can be viewed at a particular instance of time. Animated view can also be obtained.

The user can select the fixed (ground) reference frame, or view the model from a reference frame attached to one of the bodies.

Different bodies in the model can be selectively turned translucent, invisible, or visible by clicking on them in a bodies tree (Figure 3.55). The tree can be expanded and the user can selectively turn on and off individual parts of a body.

The attribute tab of *iGlass* (Figure 3.56) can be used to select the attribute that is used to color the model. The attributes could be a stress resultant such as the Von Mises Octahedral stress, the maximum principal normal stress, or the minimum principal normal stress. The stress error estimate is another attribute that can be used to color the model. The color scale is also shown on the attributes tab.

The individual color and stress levels for the colors can be individually changed by the user by double-clicking on a color and/or stress value. The background color can be changed. The value of an attribute at a particular vertex of the model can be obtained by double-clicking on the vertex. The contact pressure (Figure 3.57), load distribution and bearing forces and moments

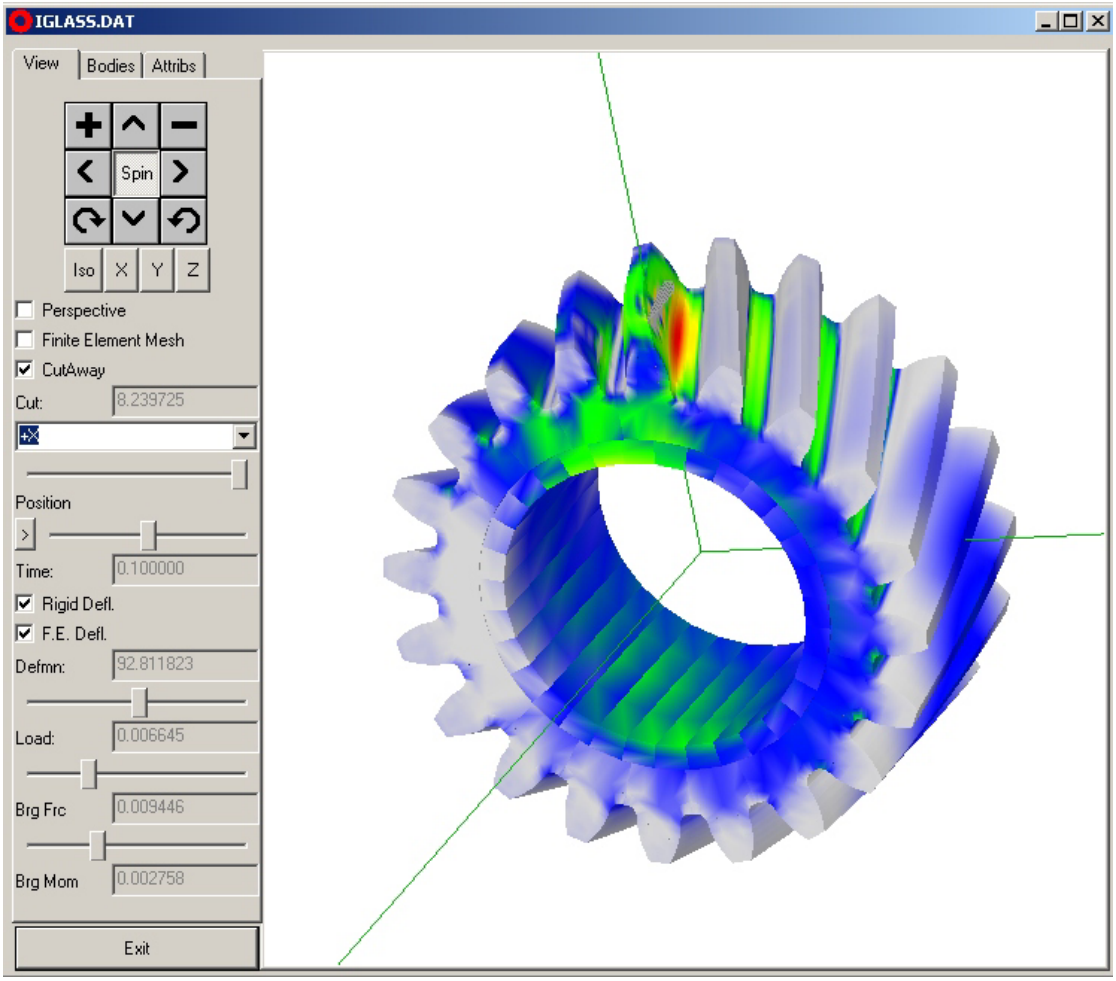

Figure 3.53: An example of an iGlass postprocessing window.

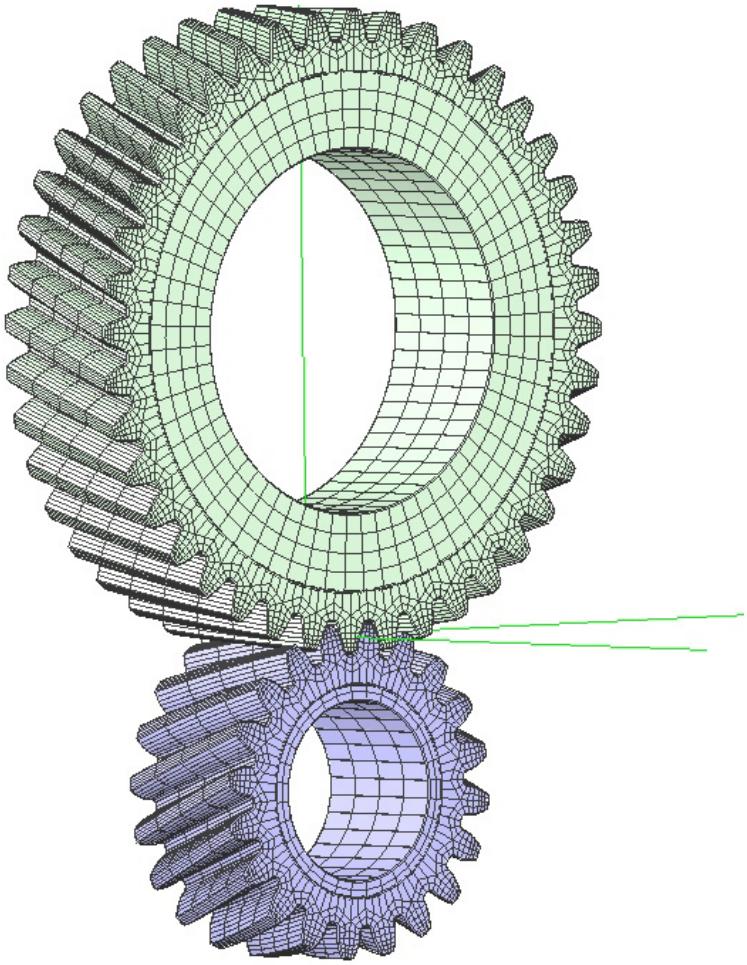

Figure 3.54: Finite element mesh model of the gear bodies

Table 3.3: Common buttons

Button	Purpose
	Select the minimum allowable value
	Decrement the value by 1
	Select the default value
	Increment the value by 1
	Select the maximum allowable value
	Accept the value just typed in
	Discard the value just typed in
	Get additional information
	Change the current graphics page
	Change the zoom level
	Refresh the graphics page

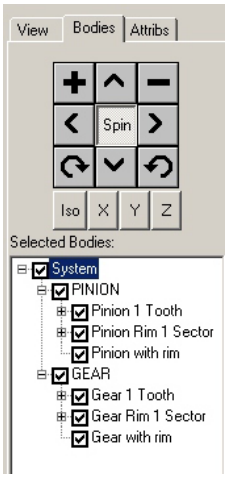

Figure 3.55: iGlass preprocessing Bodies menu

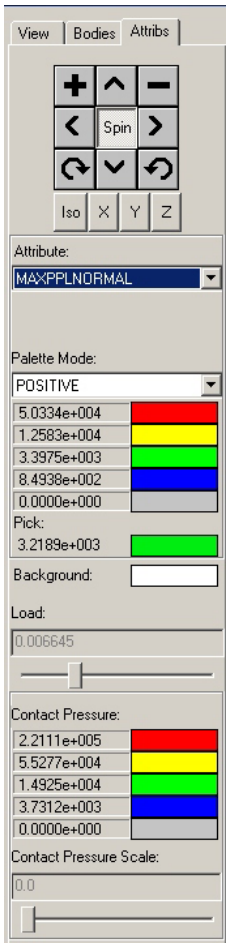

Figure 3.56: The iGlass postprocessing attribute menu.

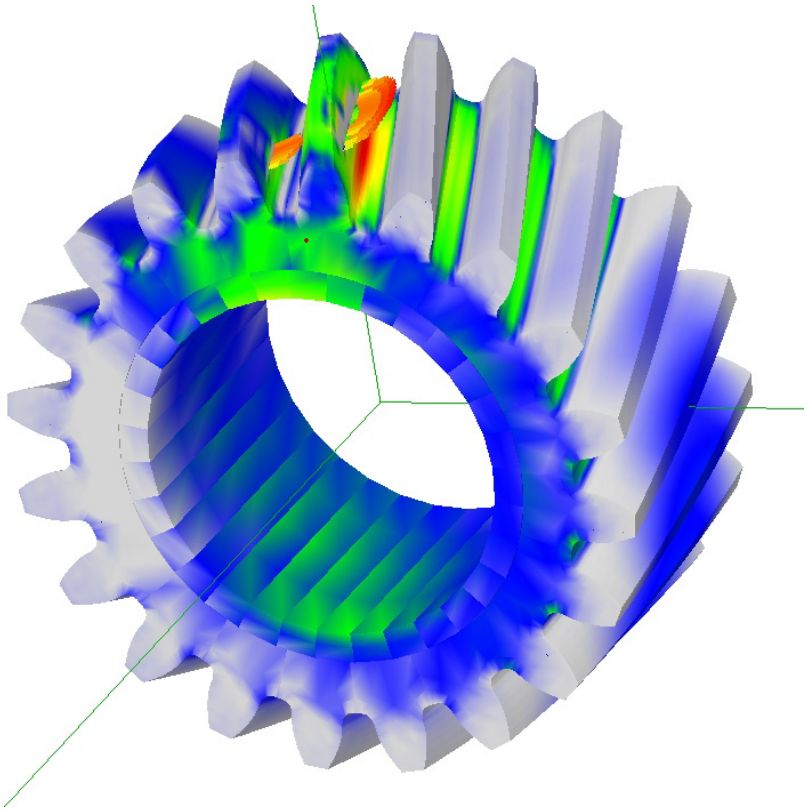

Figure 3.57: Contact pressure variation on a gear tooth

can also be displayed graphically.

Chapter 4

Planetary2D

4.1 The *Planetary2D* Module

4.2 Comparison of stress predictions with strain gage measurements

Krantz [20] has published stress measurements made by strain gages on the ring gear and sun gear of a spur planetary gear set from a helicopter transmission. We decided to build a model of the same gear set and compute root stresses using *Planetary2D*.

Figure 4.1 shows the model that was generated. The carrier is rigid and is supported by flexible bearings. There are four pinions. Each pinion is connected to the carrier through a bearing. The sun gear is also supported by bearings. The inner diameter of the sun gear is rigid and the inner diameters of the pinions are flexible. The outer diameter of the ring gear is also flexible. The ring gear is supported through splines on its outer diameter. Contact constraints are enforced between the two sides of each spline tooth and the housing. One side of the spline tooth has a clearance that is based on the amount of radial movement allowed to the actual ring gear. Many different ways of constraining the ring gear were attempted, and it was determined that this method was the most realistic, and it gave the best ring gear stress results.

The gear set was run for 600 time steps. Figure 4.2 shows the loads acting on the sun gear at this position, and the contours of maximum principal normal stress. Figure 4.3 shows the loads acting on the ring gear and the contours of maximum principal normal stress in the ring gear and a pinion. It is interesting to note how the ring gear shows significant root stress even on teeth that are not in contact, due to the rim bending effect. The pinion also shows this effect, but to a lesser extent. Figures 4.4 and 4.5 show the maximum and minimum principal normal stress variation, respectively, in the entire planetary system. Figure 4.6 is a close up of the minimum principal normal stress variation near one of the splines on the ring gear's outer diameter.

Figures 4.7 through 4.14 show the published strain gage results and the stresses predicted by *Planetary2D* at the strain gage locations as a function of time. The gages are mounted on the loaded and unloaded sides of the ring gear and sun gear teeth. The agreement between the experiment and numerical prediction is remarkable.

4.3 Dynamic Analysis

4.3.1 Introduction

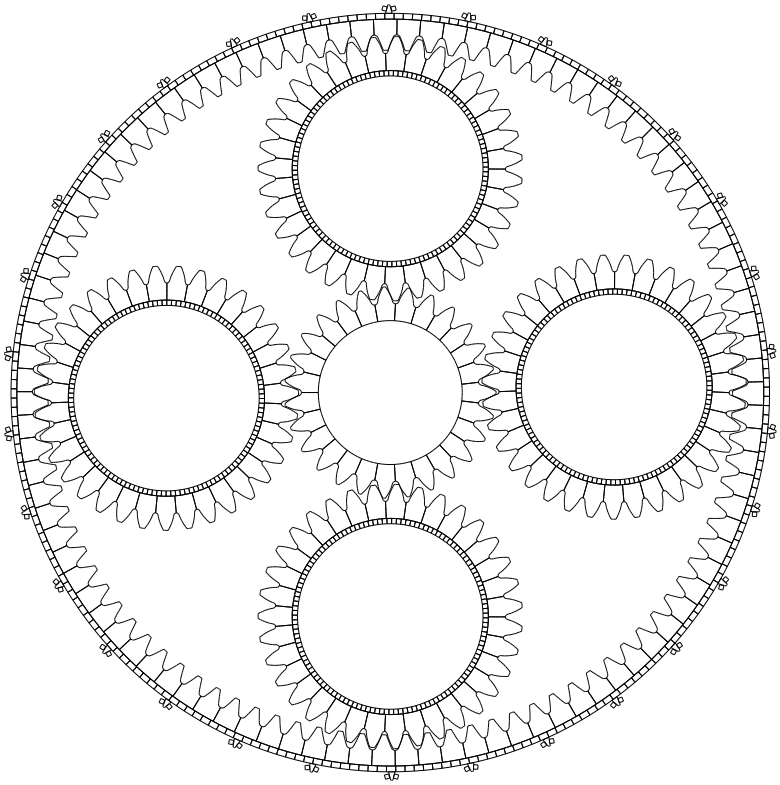

Figure 4.1: Planetary gear set model created through *Planetary2D* for comparing results.

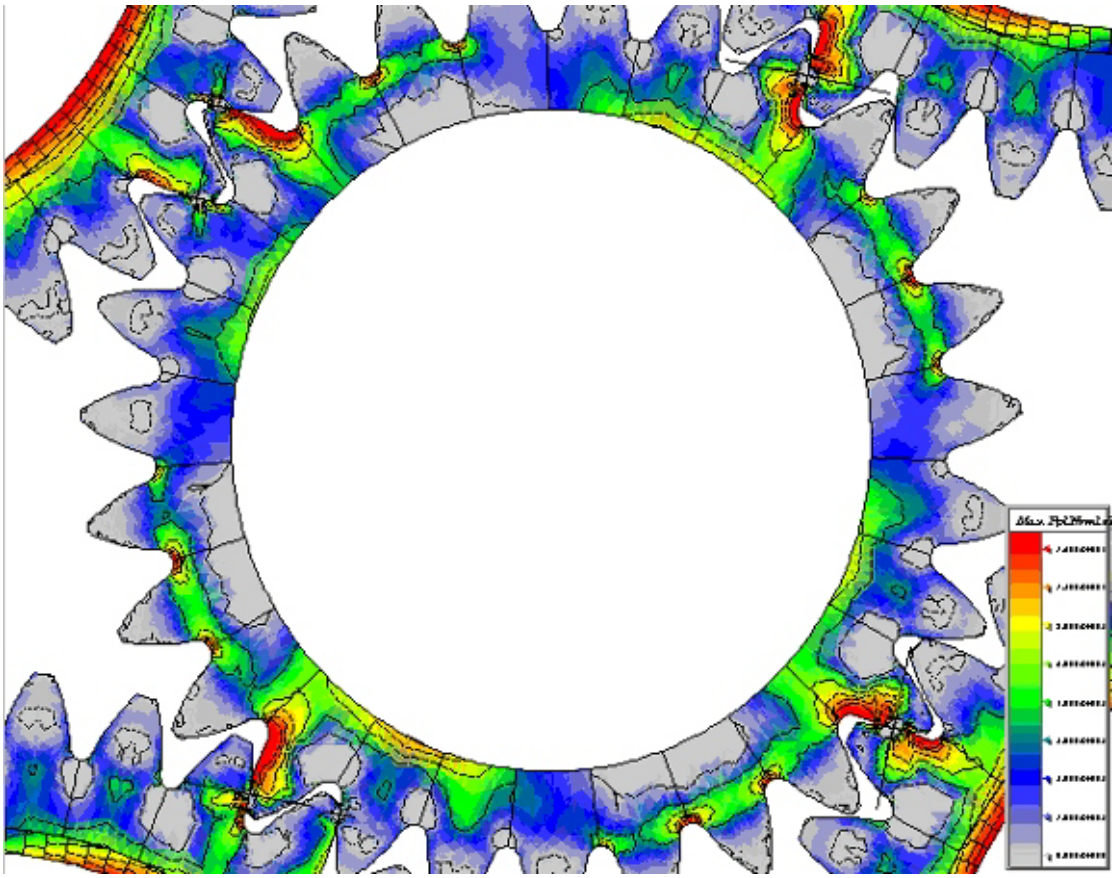

Figure 4.2: Contours of maximum principal normal stress in the sun gear.

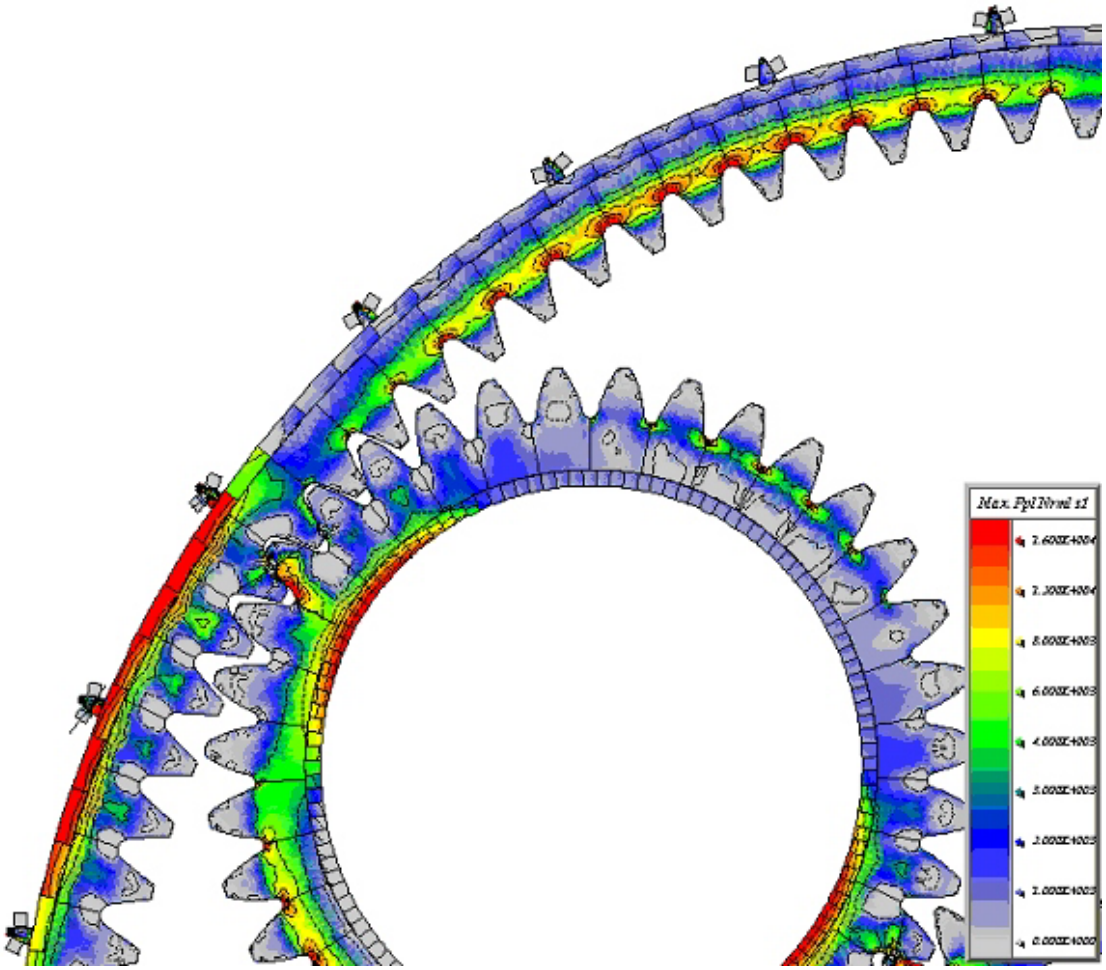

Figure 4.3: Contours of maximum principal normal stress in the ring gear and the pinion.

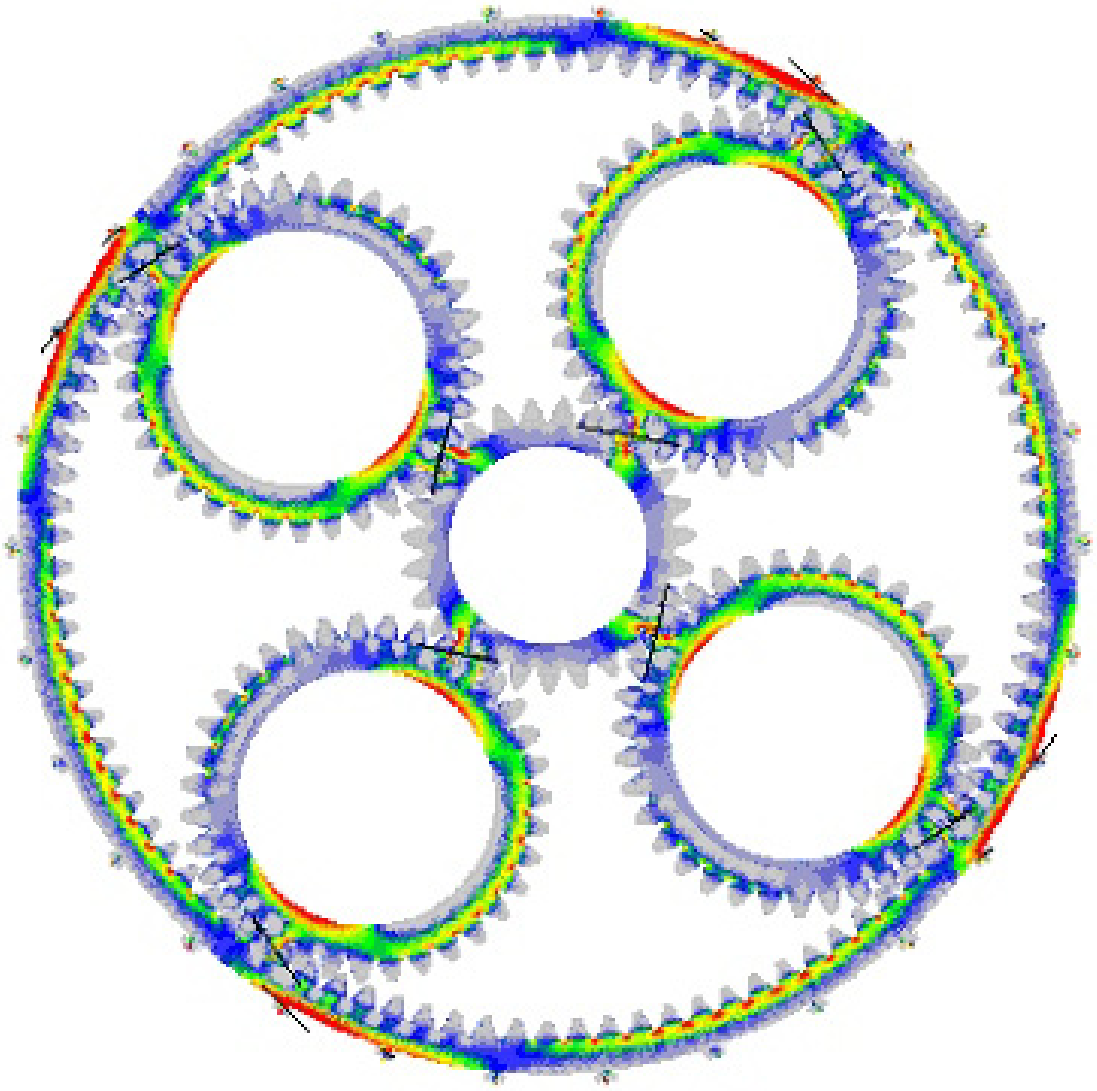

Figure 4.4: Variation of maximum principal normal stress in the planetary gear system.

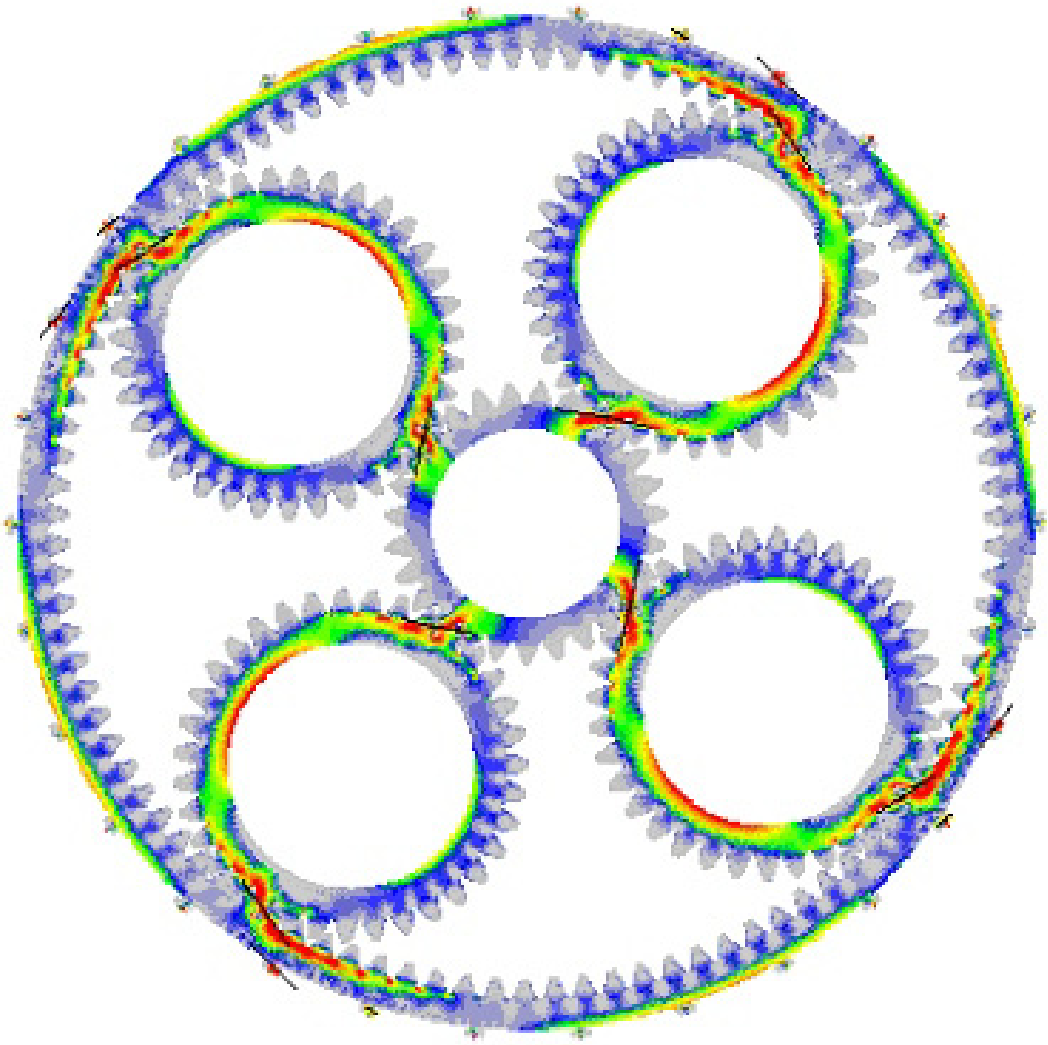

Figure 4.5: Variation of minimum principal normal stress in the planetary gear system.

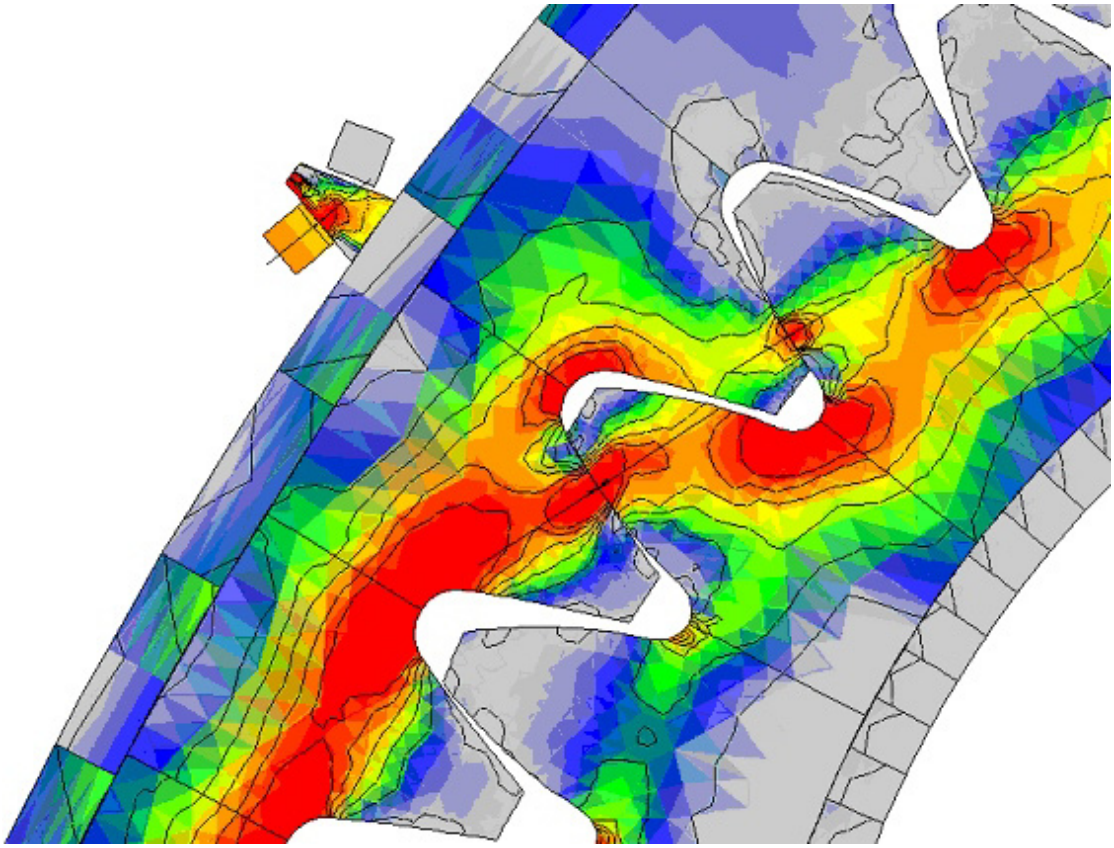

Figure 4.6: Closeup of the minimum principal normal stress variation near one of the splines on the ring gear's outer diameter.

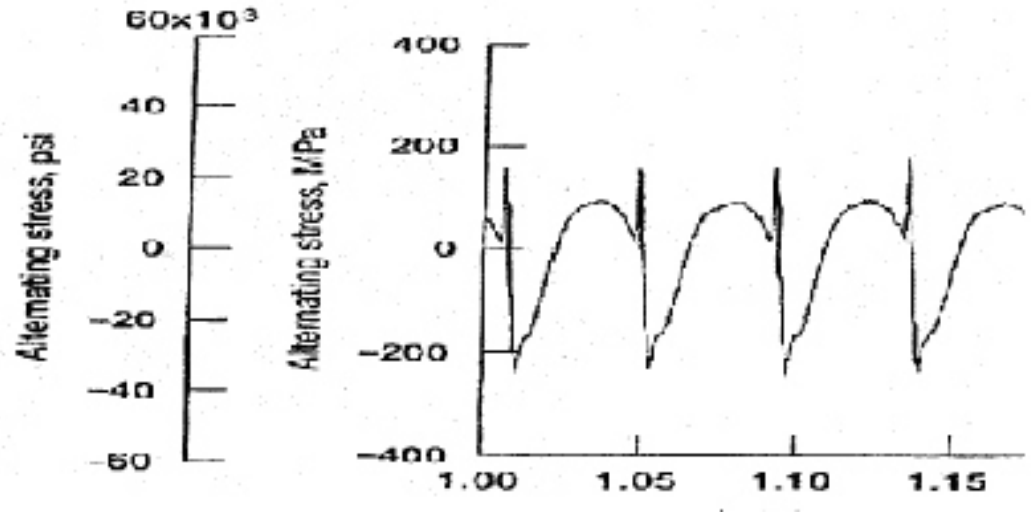

Figure 4.7: Experimental results for stress obtained from strain gages mounted in the fillet on the loaded side of the ring gear tooth. Experimental data is from Krantz [20].

Figure 4.8: Computational predictions of fillet stress on the loaded side of the ring gear tooth.

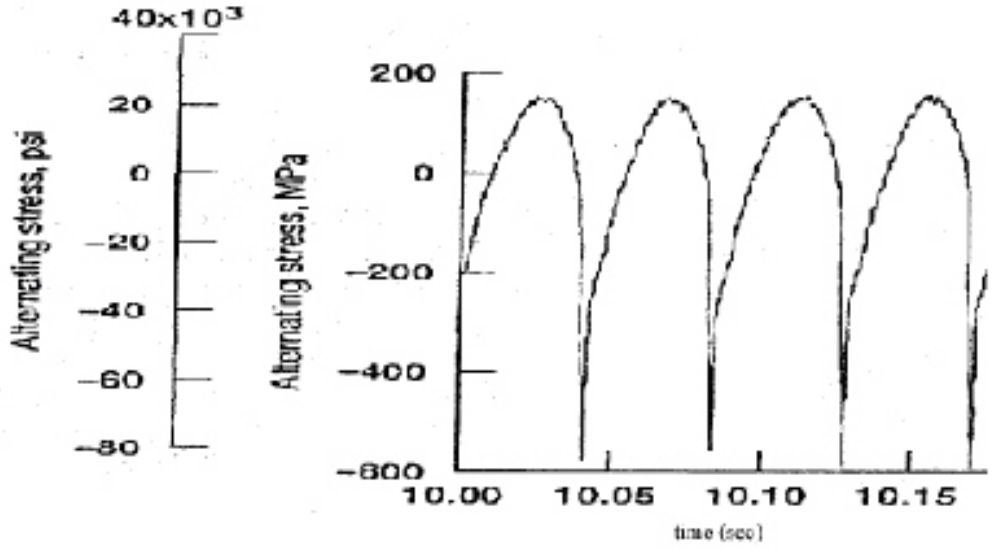

Figure 4.9: Experimental results for stress obtained from strain gages mounted in the fillet on the unloaded side of the ring gear tooth. Experimental data is from Krantz [20]

Figure 4.10: Computational predictions of fillet stress on the unloaded side of the ring gear tooth.

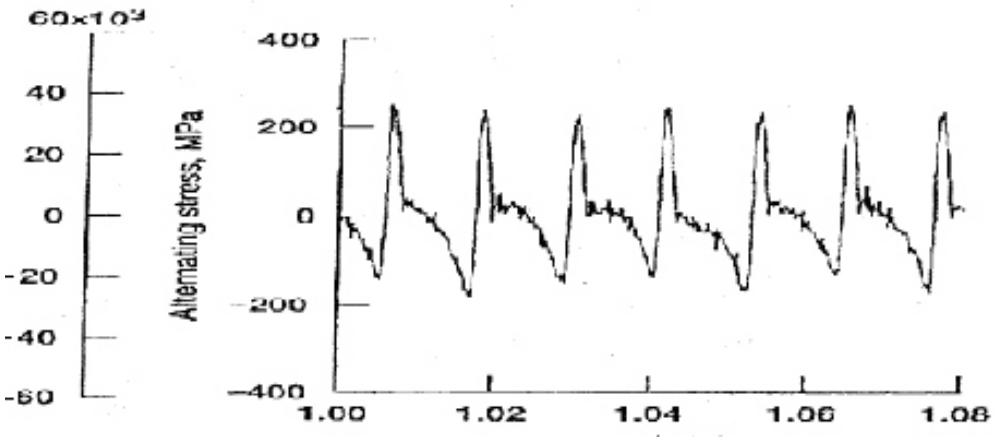

Figure 4.11: Experimental results for stress obtained from strain gages mounted in the fillet on the loaded side of the sun gear tooth. Experimental data is from Krantz [20].

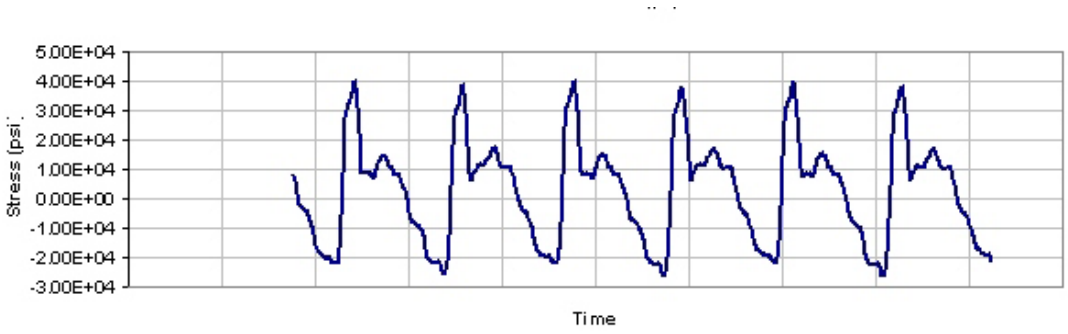

Figure 4.12: Computational predictions of fillet stress on the loaded side of the sun gear tooth.

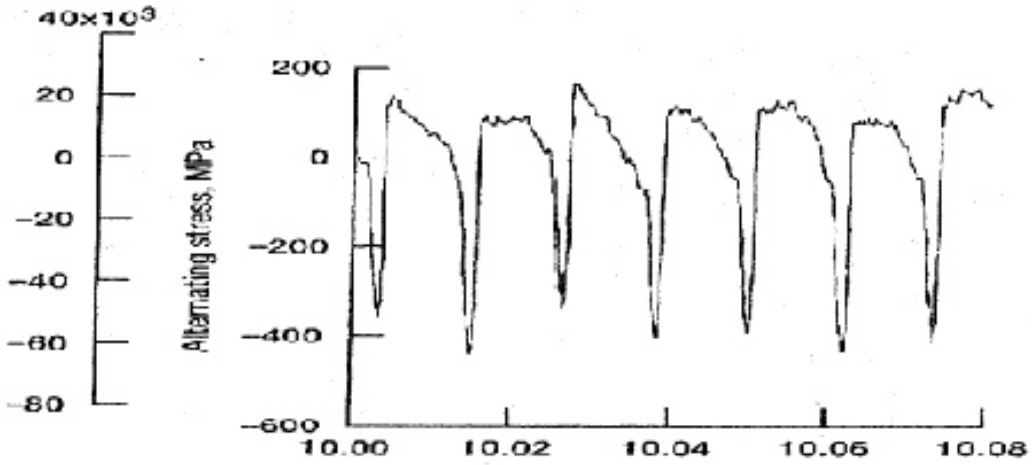

Figure 4.13: Experimental results for stress obtained from strain gages mounted in the fillet on the unloaded side of the ring gear tooth. Experimental data is from Krantz [20].

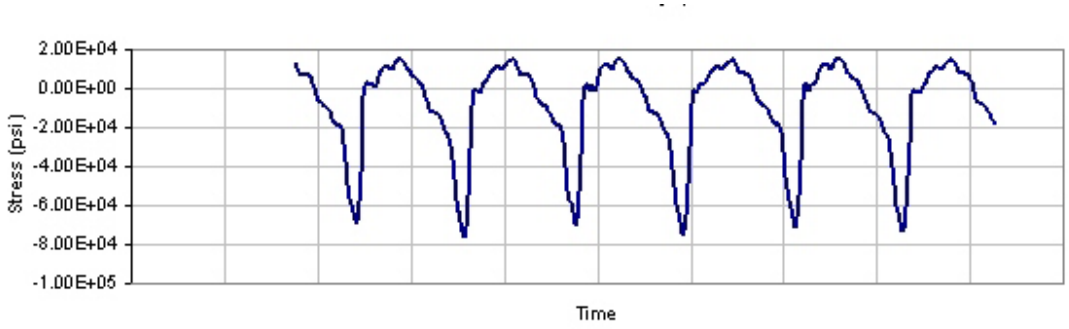

Figure 4.14: Computational predictions of fillet stress on the unloaded side of the ring gear tooth.

In this chapter, the results of dynamic analysis from Planetary2D are validated with results from analytical (lumped parameter) models, where available. The system considered is the last stage reduction unit of a helicopter transmission (this example is part of standard Planetary2D installation). In this configuration, the ring gear is fixed and the sun gear and the carrier are the input and output members respectively.

Two different models are considered for analysis. In the first model, the ring gear is press fitted to a rigid housing, while in the second model, the ring gear is attached to the housing through splines (lugs). The model with splines provides accurate contact boundary conditions, and the stress levels predicted by this model match with measured data as reported in an earlier chapter. Since analytical solutions do not exist for the second model, results from Planetary2D dynamic analysis are shown along with arguments justifying the observed results.

4.3.2 Free Vibration Modes

The free vibration modes of the planetary system are computed from an idealized lumped parameter model where in gears are rigid bodies interconnected by springs representing mesh stiffness and support bearings. The Matlab program `planet_modes.m` enables the user to compute the natural frequencies and modes. The GUI interface in Figure 4.15 allows for easy input of inertial properties, bearing/tooth stiffness and boundary conditions. To derive the equations of motion, it is necessary to transform the the body motions to a coordinate system with an axis along the line of contact to compute the spring deformation. The spring force along the line of action is then resolved back to the body coordinate system to yield the equations of motion. Mode shape animation helps with the visualization. The user is referred to [28] for details of the modeling and equations of motion. Here we briefly describe the important results.

The cyclic symmetry of planetary gears leads to repeated modes. There are at most *fifteen different natural frequencies* when the number of planets $N \geq 3$. Table 4.1 lists the natural frequencies and their multiplicity. Some interesting conclusions can be drawn from Table 4.1 and the associated mode shapes.

Table 4.1: Natural frequencies (Hz). $N = 4$ and free-free boundary conditions.

Multiplicity, m	Frequency, Hz
1	0
	1536
	1971
	2626
	7774
	13072
2	727
	1091
	1893
	2343
	7190
	10438
$N-3$	1808
	5964
	6982

1. Six natural frequencies always have multiplicity $m = 1$ for any N. The vibration modes have *pure rotation* of sun, carrier and ring and are called *rotational modes*. *All planets*

have the same motion and move in phase. In rotational modes, the tooth mesh forces are in phase. Due the symmetric geometry, the net force on sun, carrier and ring is zero, but the net moment is non-zero causing a rotational motion.

2. Six natural frequencies always have multiplicity $m = 2$ for any N. These vibration modes have *pure translation* of sun, carrier and ring and are called *translational modes*. In translational modes, *opposite planets move out-of-phase* (for N even). Here, the net moment on the sun, carrier, and the ring is zero, but the total force is non-zero.

3. The remaining modes have multiplicity $N - 3$. The vibration modes are termed *planet modes* since the sun, carrier and ring are stationary. Only planet motion occurs in these modes. In this case,*adjacent planets move out-of-phase* (for N even). In these modes, both total force and moment on the sun, carrier and ring is zero.

These well defined eigensolution properties are useful for to explain selective participation of different classes of vibration modes in dynamic response [29, 6].

4.3.3 Validation of Impulse Response Functions

Planetary2D directly yields the dynamic response. For the non-rotating system [ring fixed, input torque to the sun gear, output from carrier (carrier rotation fixed)], an impulse force/moment is applied to the sun bearing for a single time step. The time domain impulse response of carrier, ring and planets are computed using Planetary2D (use the session files `ImpulseResponseSunFx.ses` and `ImpulseResponseSunMz.ses` for these simulations). The time domain impulse response functions are converted into frequency domain transfer function via FFT techniques.

The natural frequencies of the participating modes are computed from the peaks in the transfer functions. Matlab program `fftresponse.m` can be used to plot the transfer functions and visualize the operating deflection shapes at any selected frequency. These natural frequency estimates are compared with the exact solutions from the lumped parameter model. In addition, Planetary2D transfer functions are compared with the responses from analytical modal superposition.

The driving point frequency responses due to force (x-direction) is shown in Figures 4.16. There is excellent agreement between the Planetary2D solution which computes the transient solution due to the impact excitation and the solution from the lumped parameter model, where the solution is computed in the frequency domain using modal superposition.

The natural frequency comparisons are shown in Table 4.2. In general, there is good agreement between Planetary2D and the analytical model. The natural frequency predictions of the lumped parameter model seem to be a bit on the higher side.

Table 4.2: Natural frequency comparison for the four planet gear system.

Natural Frequency, Hz	Planetary2D	Lumped Parameter Model	Difference (%)
Translation	803	825	-2.6
	1722	1833	-6.0
	2181	2325	-6.2
Rotation	1150	1243	-7.5
	1778	1903	-6.5

4.3.4 Dynamic Response Under Operating Conditions

To study the dynamic response of the planetary gear system under operating conditions, several dynamic analyses were conducted over a range of input speeds from 100-4200 RPM. In the configuration analyzed, a constant input torque of 16200 lbf-in is applied to the sun gear. Each analysis computes the response at a single speed, and is subdivided into several ranges to facilitate loading the system statically, a ramp up phase to bring the system to desired speed and a final range where the system is in steady state. The time series data in the steady state range is analyzed in the frequency domain. This analysis procedure is repeated for each speed.

Planetary2D outputs the body deflections and forces in text format in files `CARRIERBRGRES.DAT`, `CARRIERRES.DAT`, `HOUSINGRES.DAT`, `PINION1BRGRES.DAT`, `PINIONnRES.DAT`, `PINIONnBRGRES.DAT`, `RINGRES.DAT`, `SUNBRGRES.DAT`, `SUNRES.DAT`. This data was post-processed in Matlab to obtain the steady state frequency response amplitudes. Matlab M-files `post_process_calyx_data.m` and `process_order_plots.m` are developed for this purpose to view the spectrograms and the order maps from the speed sweep runs.

Planetary2D also also outputs the file `INPUTTHETA.DAT`, which contains the instantaneous input angle as a function of time. The utility program `generate_tacho.m` generates a specified number of pulses for one complete rotation of the input member, which can be used to perform adaptive resampling of the time domain data for order analysis.

In the model used for speed sweep runs (Model 2), the ring gear is connected to the housing by 50 splines or lugs. The natural frequencies of the system are changed due to the different boundary conditions, as clearly seen from the results of the numerical impact tests using Planetary2D. There is also some coupling between the translational modes and the rotational modes, in the sense that the motion of the sun, carrier and ring bodies in the translational mode will have a small amount of rotation as well. The natural frequency estimates are obtained from the peaks in the computed transfer functions, which can be plotted using the Matlab program `fftresponse`. The natural frequencies are shown in Table 4.3.

Table 4.3: Natural frequency comparison for the four planet gear system.

Natural Frequency, Hz	Splined Model	Fixed Ring Model
Translation	921	803
	1895	1722
	2227	2181
	2952	
	3532	
	4298	
Rotation	1143	1150
	2421	1778
	3121	

In a planetary system, tooth forces in all the meshes are periodic with the fundamental period T_{cycle}. However there is relative phase difference between the forces in different meshes. Since the excitation is periodic, the frequency content of the excitation is predominantly concentrated at the mesh frequency and its higher harmonics. The mesh frequency is proportional to the input speed. There is an amplification in response whenever a harmonic of the tooth mesh frequency coincides with a system natural frequency. A splined model will have additional excitation at the spline passage frequency and its harmonics. In the current model, the ring gear (99 teeth) is connected to the housing through 50 splines. At a given speed, the fundamental spline pass frequency will be approximately one-half (50/99) of that of the tooth mesh frequency.

4.3.5 Results and Discussion

The spectrograms and the order colormaps for the translational and rotational deflections of the sun are shown in Figures 4.17 and 4.18. Several interesting physical phenomena can be observed from these figures.

From Figure 4.17, it is clear that only the odd harmonics of the mesh frequency are dominant in the sun translation response. At frequencies of 943 Hz and 2206 Hz corresponding to the first and third translational modes, there is an amplification if there is a coincidence with odd tooth mesh harmonic. These peaks in dynamic response correlate very well with the natural frequency prediction from the system impulse response shown in Table 4.3. The spline pass harmonics are (at roughly about half the tooth mesh frequency) evident in the response as well.

Similarly, in Figure 4.18 only the even harmonics of the tooth mesh frequency are dominant. Again, there is excellent correlation between the peaks in the dynamic response (at 1169 Hz and 2349 Hz, corresponding to the first and second rotational modes) and the natural frequencies predicted from numerical impact test. The spline pass harmonics are seen in the rotational response as well.

This peculiar behavior where particular modes are excited in some mesh harmonics but not others is consistent with analytical predictions and can be explained by a practice called *planet phasing*. A parameter k is defined as $k =$ mod (lZ_s/N), where l denotes the harmonic of the tooth mesh frequency, Z_s is the number of teeth on the sun gear and N denotes the number of planets. The planets in the four planet system are not equally spaced but lie along diameters. For even values of lZ_s translational modes are not excited and rotational modes are excited. For odd values of lZ_s, rotational modes are not excited, but translational modes are excited. Hence the results predicted by Planetary2D verify the theoretical predictions [29]. This knowledge is useful in the design stage to suppress either translational or rotational modes, by intentional phasing between the planets.

Figures 4.19 and 4.20 show the radial and rotational motions of one of the planets. All the tooth mesh harmonics and the spline passage harmonics participate in the planet motions, since for an individual planet participating in only two gear meshes the forces and moments do not cancel, unlike the sun gear.

The flexibility of the ring gear introduces additional dynamics at low frequencies. If the ring gear has uniform mass and stiffness, it can be shown that the exact shape of the mode of vibration consists of a sinusoid along the ring circumference [35]. The exact formula for the natural frequencies is given as

$$\omega_n = \frac{n(n^2-1)}{\sqrt{n^2+1}}\sqrt{\frac{EI}{\mu_1 R^4}} \tag{4.1}$$

where $n > 1$ is the number of full waves, μ_1 is mass per unit length of the ring, EI is the bending stiffness and R is the ring radius. The mass per unit length of the ring gear is computed from Planetary2D by querying the total mass of the ring gear and dividing by its circumference. An effective cross-section moment of inertia is computed from the ring gear dimensions. The geometry parameters of the ring gear are shown in Table 4.4.

Table 4.4: Geometry parameters to compute ring gear natural frequencies.

Elastic Modulus, (N/m^2)	2.10E+11
Moment of Inertia of the Cross Section, (m^4)	1.21E-09
Mass Per Unit Length, (kg/m)	1.8926
Radius, (m)	0.17088

Figure 4.21 displays the ring gear translational response. It is seen that only the ring gear modes that have even number of full waves around the circumference are excited. More detailed

Table 4.5: Ring gear modes and natural frequencies.

Number of Full Waves n	Frequency, Hz
2	169.39
3	479.11
4	918.66
5	1485.68
6	2179.46
7	2999.75
8	3946.43
9	5019.45
10	6218.78

study is required to explain the the physical reason behind phenomenon. One potential reason may be our system has even number of pinions (four). There is good correlation between the predicted and computed natural frequencies.

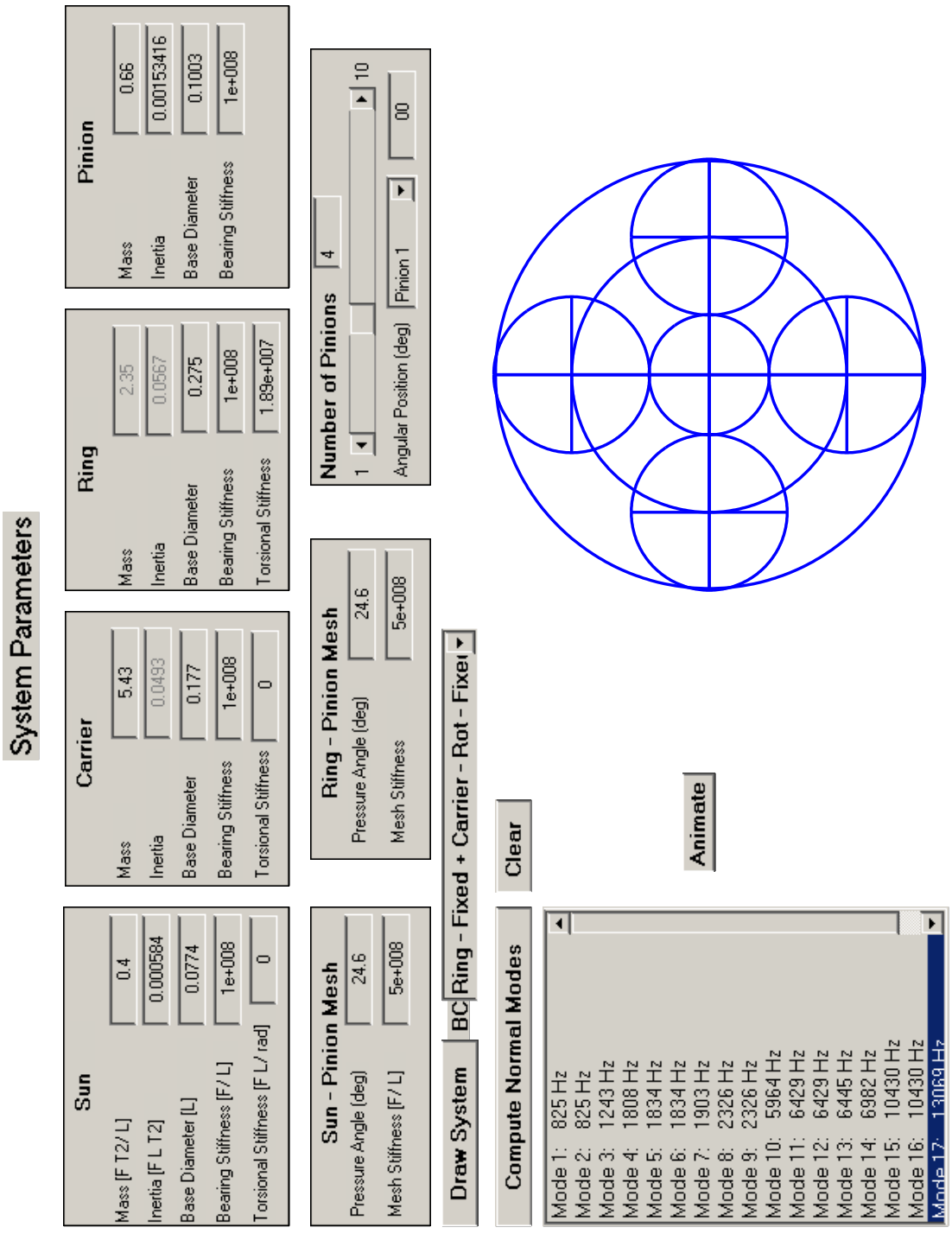

Figure 4.15: Matlab program for computing vibration modes of a planetary system.

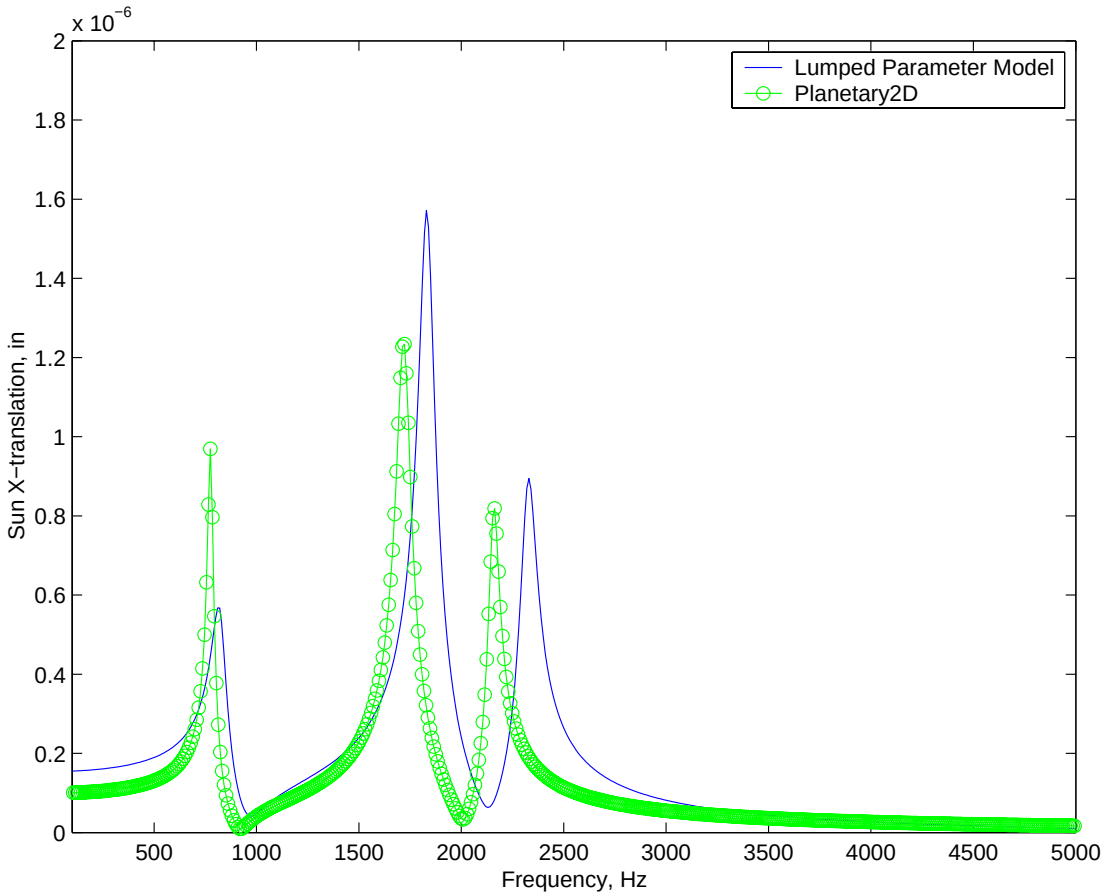

Figure 4.16: Driving point response – sun x-translation.

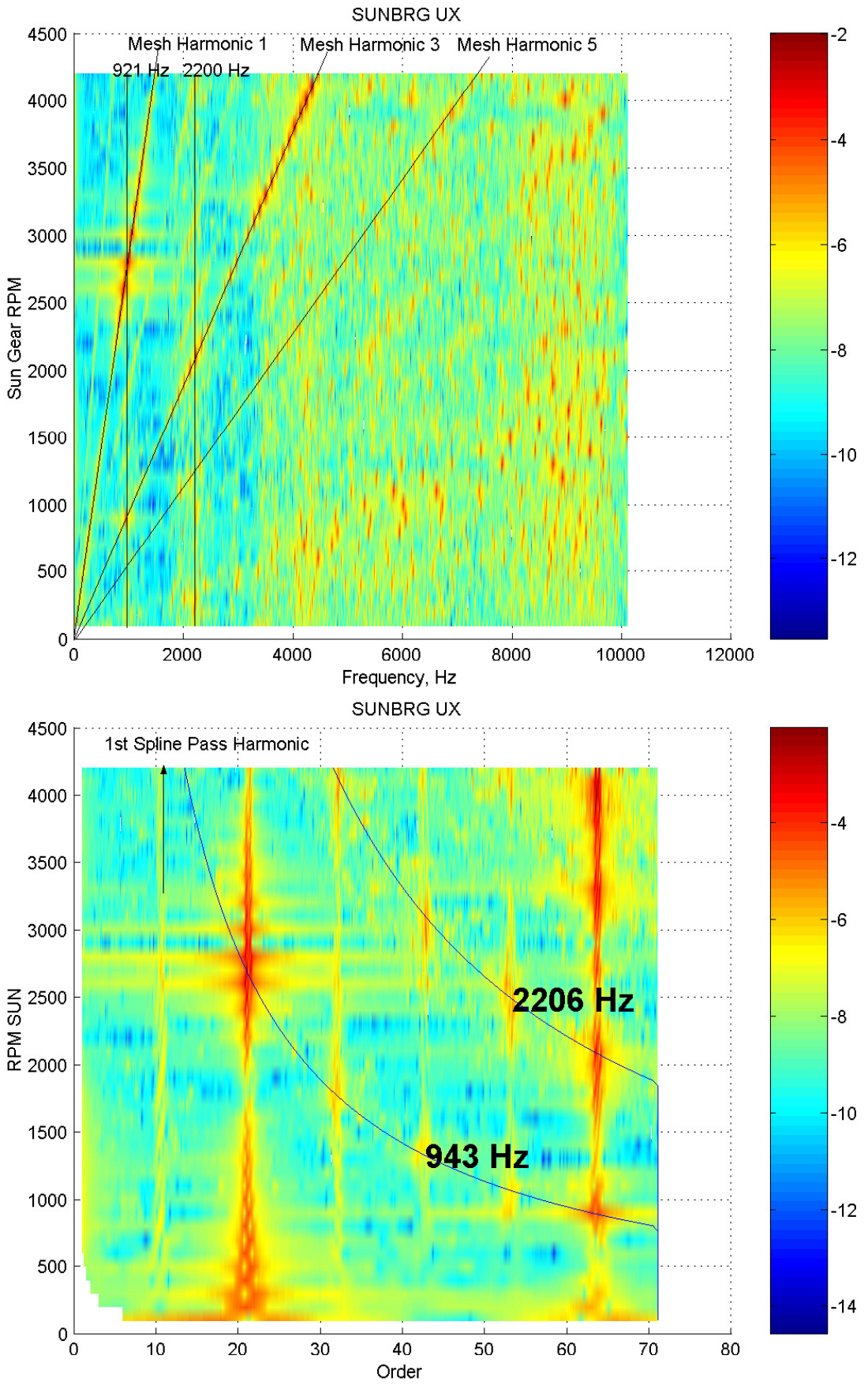

Figure 4.17: Frequency spectra of sun x-translation.

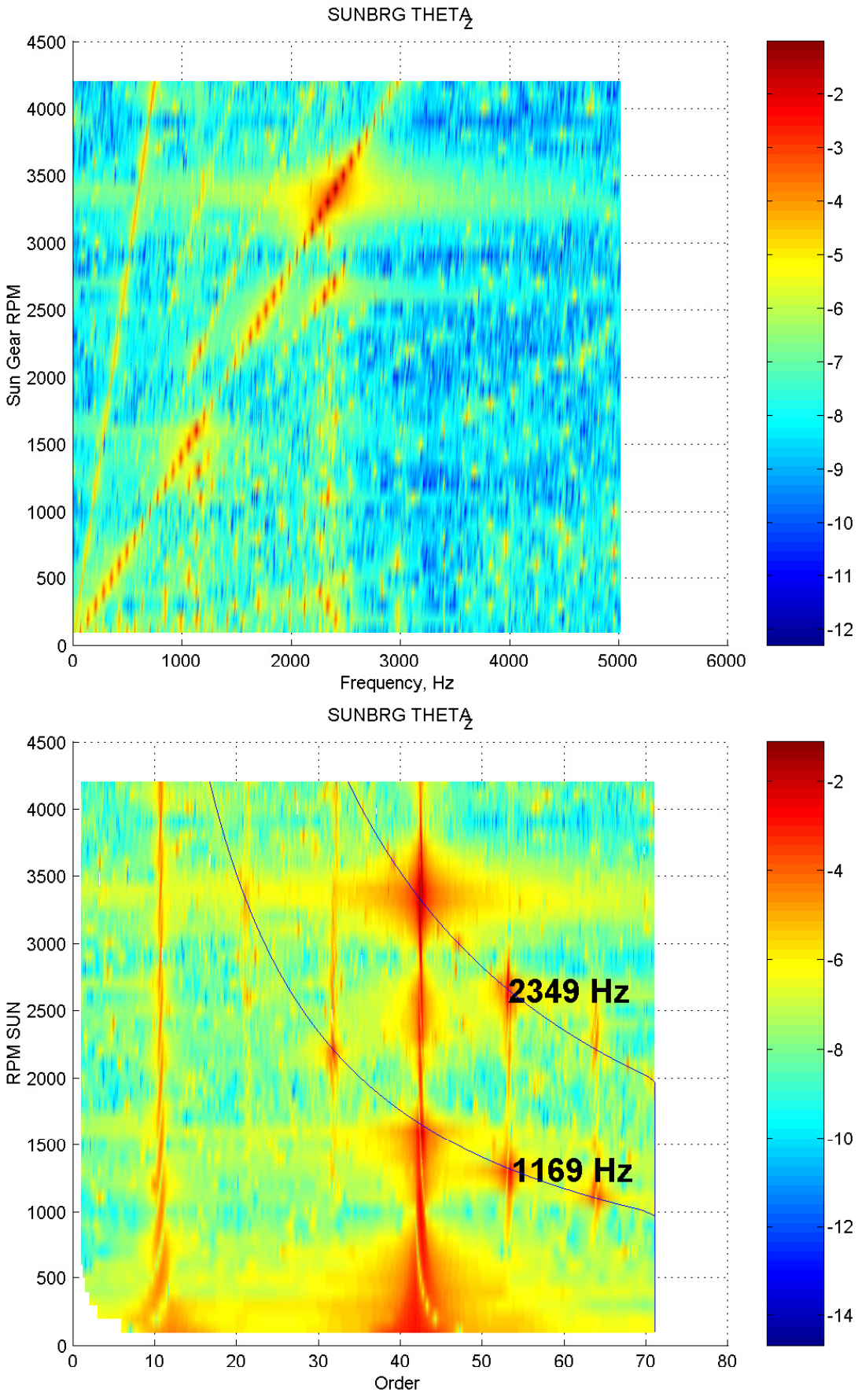

Figure 4.18: Frequency spectra of sun θ_z rotation.

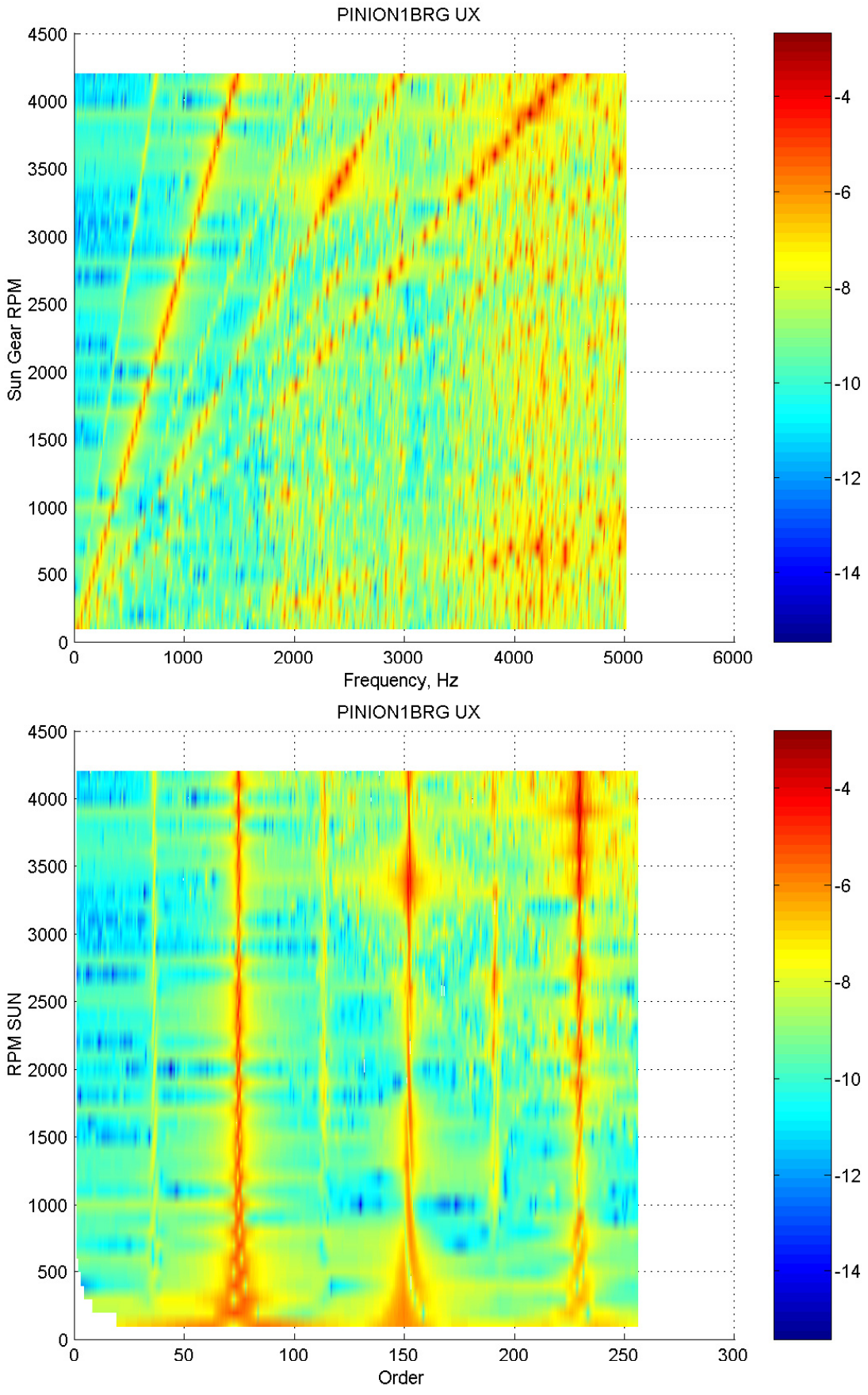

Figure 4.19: Frequency spectra of pinion radial translation.

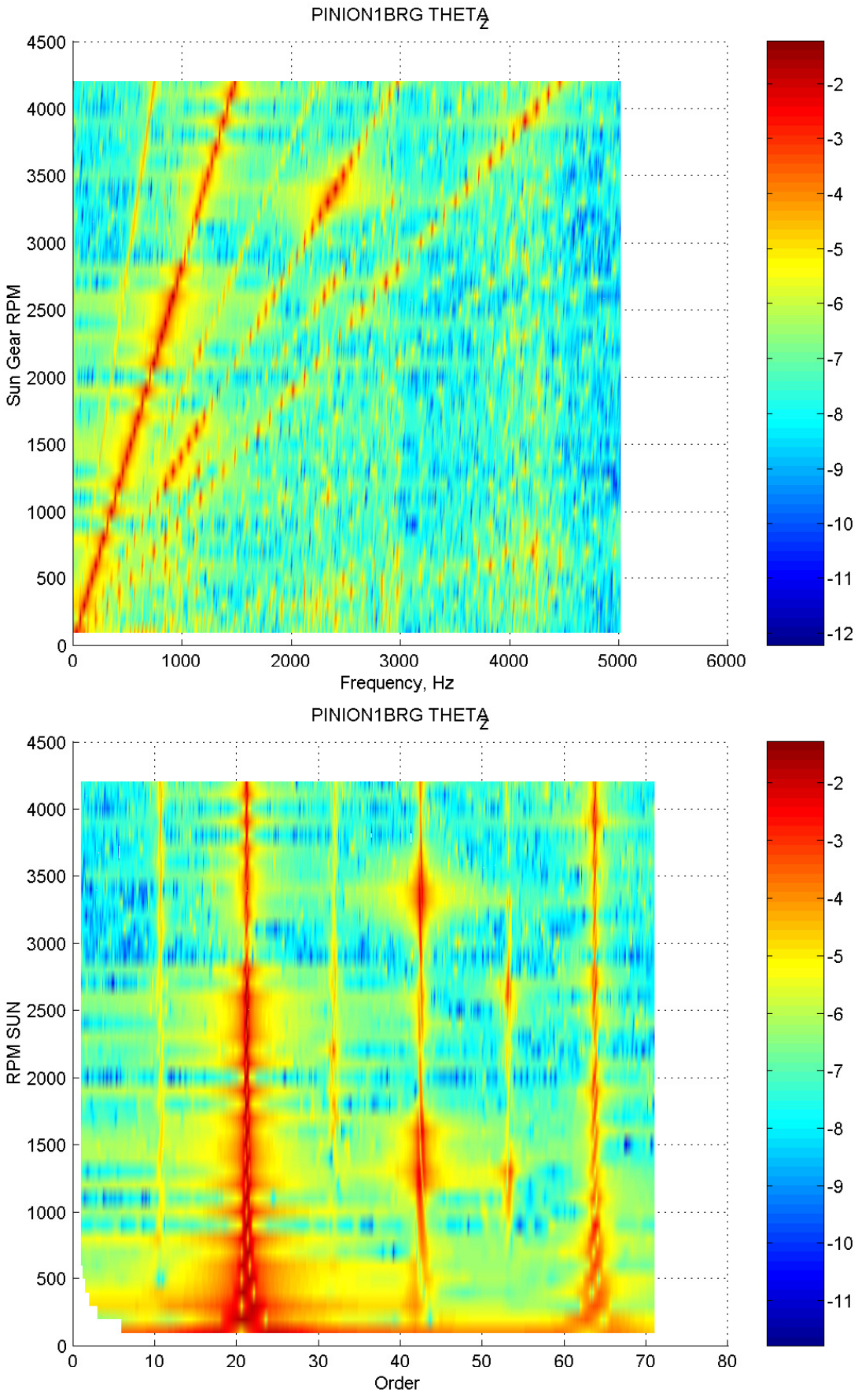

Figure 4.20: Frequency spectra of pinion rotation.

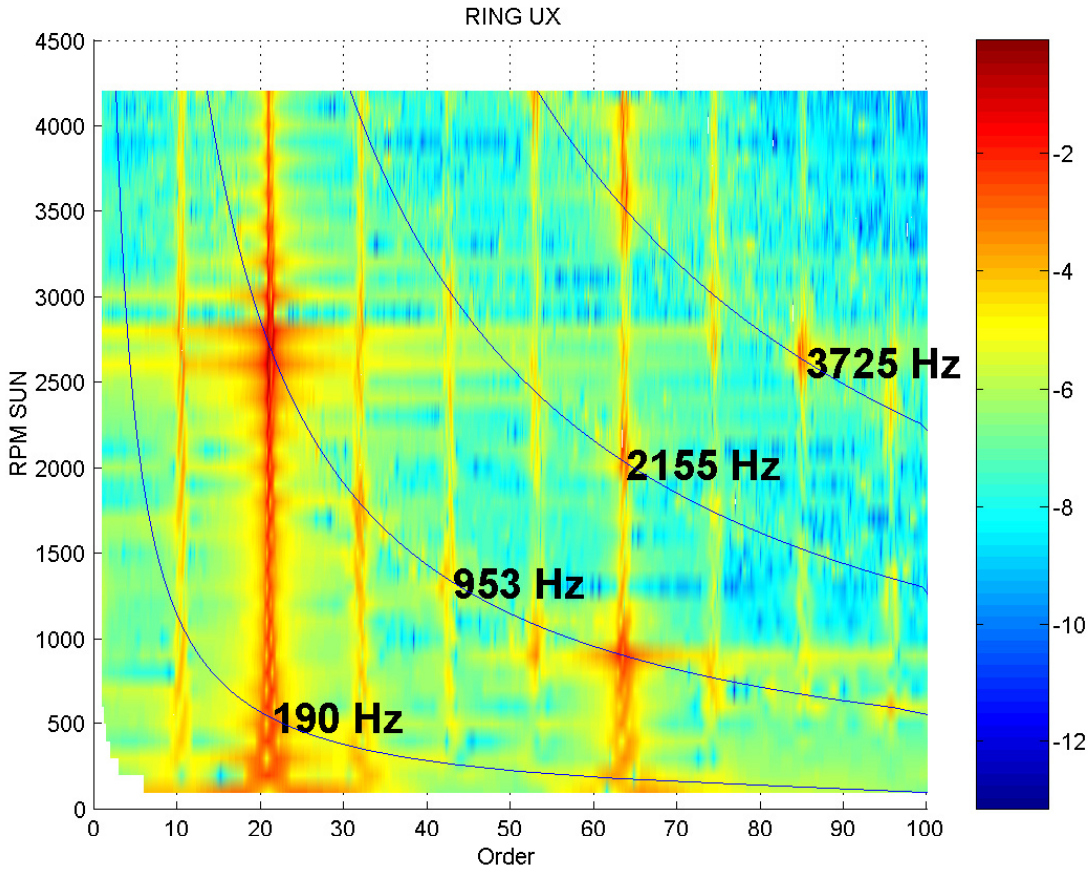

Figure 4.21: Frequency spectra of ring gear translation.

Chapter 5

Helical3D

5.1 The *Helical3D* Module

Helical3D is a software module that we have built for the analysis of external and internal spur and helical gear pairs. The module was built using *Calyx*, *Multyx* and *Guide* as sub-components. We have put this module through an intensive testing, debugging and validation effort. An example of a gear model created in *Helical3D* is shown in Figure 5.1.

A brief overview of *Helical3D* and some of the validation studies we carried out is provided here. A more in-depth description is provided in the *Helical3D User's Manual*[36] and in the *Helical3D Validation Manual*[37].

5.1.1 Basic geometry

The basic geometry of for the helical tooth form is specified through the usual gear design parameters. Three diameter values are needed, the root diameter, the outer diameter and the rim diameter. For an internal gear, the minor diameter is specified instead of the outer diameter. The user can specify the pressure angle, the diametral pitch and the thickness for a normal section of the tooth, or for a transverse cross section. The helix angle, the face width and the tip radius of the rack tooth are the other important input parameters. The user can choose to specify an arbitrary transverse cross section through a data file, instead of using the standard involute form.

The loading is provided through the input torque. The user can ask *Helical3D* to look for contact on only one side of the tooth, or on both sides in case back side contact is suspected. If back side contact is enabled, the user can choose to apply a force along the center distance direction instead of keeping the pinion and gear at a fixed center distance. This is helpful in analyzing 'tight mesh' conditions.

Two angular misalignments can also be introduced into the system.

The level of refinement of the finite element mesh is controlled by selecting a mesh 'template' (see 3.2.2 for more details).

5.1.2 Surface Modifications

Several surface modification types have been implemented. Standard linear and quadratic profile modifications can be applied to the tip and to the root. A standard parabolic crown modification can be applied in the lead direction. In addition to these standard modifications, any arbitrary profile modification can be specified in tabular form. Similarly, an arbitrary lead modification can be applied in tabular form. Finally, an arbitrary topographical modification can be specified as a two-dimensional table.

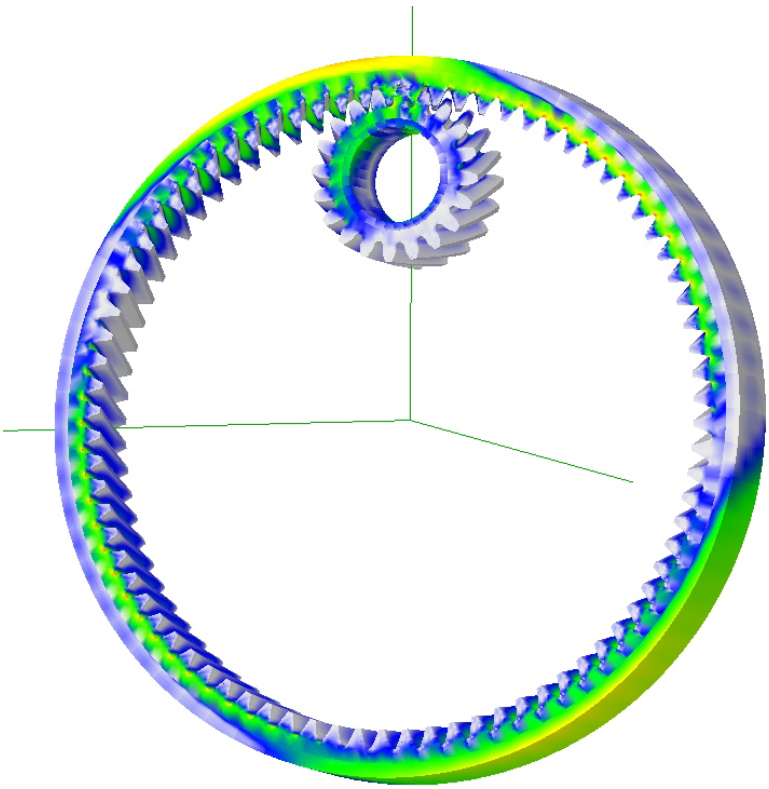

Figure 5.1: An internal gear model created with *Helical3D*.

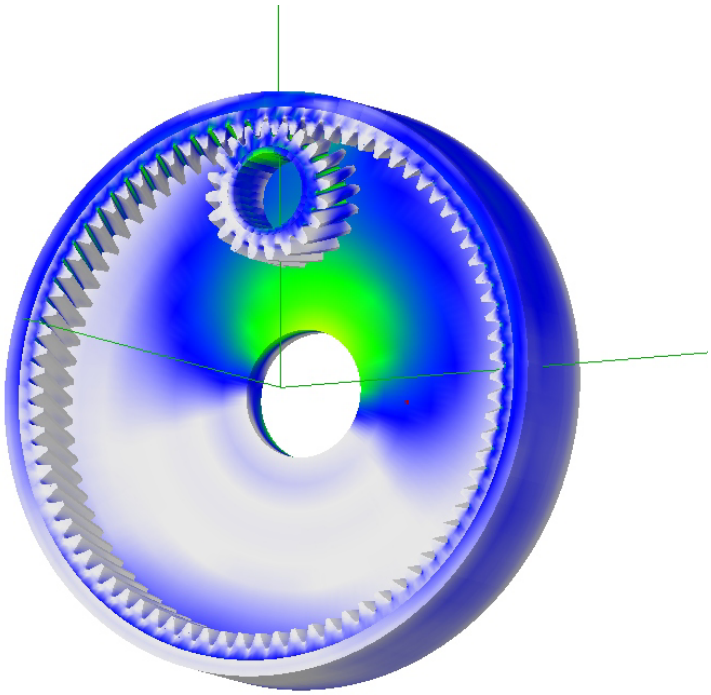

Figure 5.2: An internal helical gear with a webbed rim

5.1.3 Rim and Splines

In certain situations the boundary conditions applied to the inside of the gear can have a considerable effect on the predicted tooth bending stress and contact pressure. Two types of rim models are implemented in *Helical3D* to improve the accuracy of predictions in these situations. Figure 5.2 shows an internal gear supported by a 'Webbed' rim. *Helical3D* also allows the user to react the gear or pinion torque through splines, instead of a constrained diameter. Figure 5.3 shows an external gear supported through internal splines.

5.1.4 Bearings

Helical3D allows the use of flexible 'Bearings' to support the pinion and the gear. These can simulate the effect of flexible shafting if the effective shaft stiffness at the gear is known. The stiffness matrix for the bearing is entered into a data file, and the name of this data file is provided to *Helical3D*.

5.2 Description of the nominal system used in the numerical studies

The results of some of the validation studies we ran are presented in this chapter. A 20×30 tooth spur gear system was used for these numerical studies. Tables 5.1 through 5.7 show the important data for this system. No assembly errors were considered for the pinion and the gear. There are no bearings in the model.

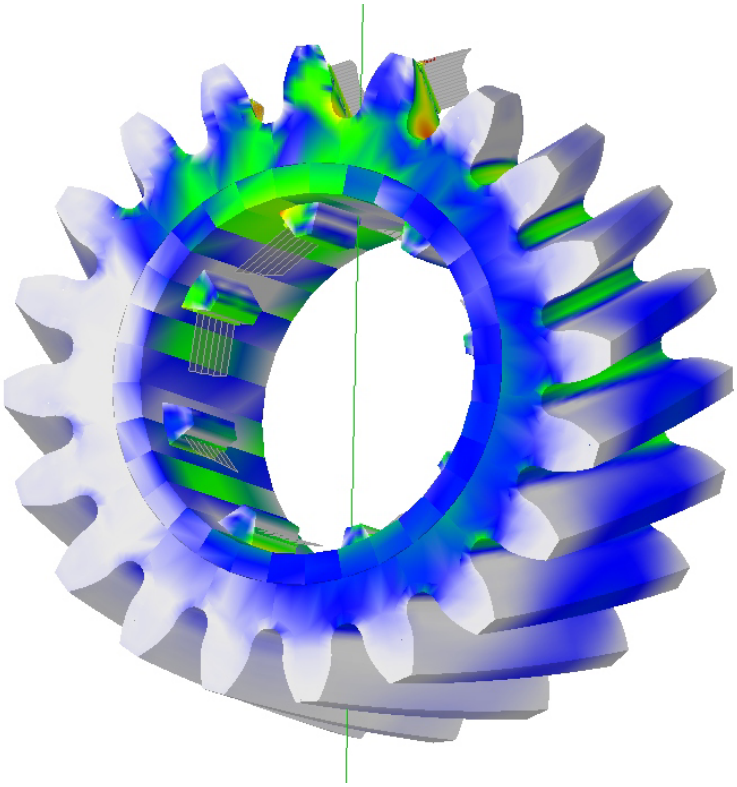

Figure 5.3: An external gear with splines

Table 5.1: System configuration parameters

Item	Description
MESHTYPE	CALYX3D
CENTERDIST	3.00
OFFSET	0.00
ROTX	0.00
ROTY	0.00
INPUT	PINION
TORQUEINPUT	1000.00
RPMINPUT	-3.00
MU	0.00
MAGRUNOUTGEAR	0.00
ANGRUNOUTGEAR	0.00
MAGRUNOUTPINION	0.00
ANGRUNOUTPINION	0.00
BACKSIDECONTACT	FALSE

Table 5.2: Pinion data

Item	Description
LUMPMASS	0.00
LUMPMOMINERTIA	0.00
LUMPALPHA	0.00

Table 5.3: Pinion tooth data

Item	Description
NTEETH	20
NFACEELEMS	4
COORDORDER	10
DISPLORDER	3
PLANE	TRANSVERSE
XVERSEDIAMPITCH	10
XVERSEPRESSANGLE	20
XVERSETHICK	0.15708
FACEWIDTH	1
HAND	LEFT
HELIXANGLE	0.00
RACKTIPRAD	0.02
OUTERDIA	2.18
ROOTDIA	1.76
RIMDIA	1.40
YOUNGSMOD	$3x10^7$
POISSON	0.3
MSHFILE	pinion.msh
TPLFILE	medium.tpl

Table 5.4: Pinion rim data

Item	Description
RIMTYPE	SIMPLE
RIMDIA	1.40
INNERDIA	1.20
WIDTH	1.00
OFFSET	0.00
AXIALORDER	2
CIRCORDER	8
ELEMTYPE	LINEAR
NDIVSRADIAL	2
NTHETA	32
NDIVSWIDTH	4

Table 5.5: Gear data

Item	Description
TYPE	EXTERNAL
LUMPMASS	0.00
LUMPMOMINERTIA	0.00
LUMPALPHA	0.00

Table 5.6: Gear tooth data

Item	Description
NTEETH	40
NFACEELEMS	4
COORDORDER	10
DISPLORDER	3
PLANE	TRANSVERSE
XVERSEDIAMPITCH	10
XVERSEPRESSANGLE	20
XVERSETHICK	0.15708
FACEWIDTH	1
HAND	RIGHT
HELIXANGLE	0.00
RACKTIPRAD	0.02
OUTERDIA	4.18
ROOTDIA	3.78
RIMDIA	3.40
YOUNGSMOD	$3x10^7$
POISSON	0.3
MSHFILE	gear.msh
TPLFILE	medium.tpl

Table 5.7: Gear rim data

Item	Description
RIMTYPE	SIMPLE
RIMDIA	3.40
INNERDIA	2.40
WIDTH	1.00
OFFSET	0.00
AXIALORDER	2
CIRCORDER	16
ELEMTYPE	QUADRATIC
NDIVSRADIAL	4
NTHETA	64
NDIVSWIDTH	4

Table 5.8: Modification menu for the pinion tooth

Item	Description
QUADTIPMOD	TRUE
ROLLQUADTIPMOD	27.25
MAGQUADTIPMOD	0.0005
LEADCROWN	0.0005

Table 5.9: Modification menu for the gear tooth

Item	Description
QUADTIPMOD	TRUE
ROLLQUADTIPMOD	19.60
MAGQUADTIPMOD	0.0005
LEADCROWN	0.0005

5.3 Comparison of contact pressure results

For spur gears, we can easily calculate the contact pressure using Hertz formula assuming a uniform distribution of load across the face width. This can be compared with the predictions made by *Helical3D*. Figure 5.4 shows this comparison for the baseline spur gear set with no tooth modification. The contact pressure is plotted against time. The agreement is very good.

Figure 5.5 shows a similar plot when a linear profile modification of magnitude $0.0005in$ is applied to the pinion and gear teeth. Again, the agreement is excellent except at the instant when the point at the start of the tip relied makes contact. The radius of curvature is locally very high here, and *Helical3D* was able to capture its effect. Hertzes formula cannot account for the irregular contact at this point.

Figure 5.6 shows the contact pressure for a case when quadratic tip relief of magnitude $0.0005in$ was applied to the pinion and gear. Agreement between Hertzes formula and *Helical3D* is excellent.

We also ran comparisons for a case where crowning of magnitude $0.0003in$ was applied to the pinion and gear teeth in this example. Profile modifications were removed. Using formulae for elliptical contact between half-spaces with constant surface curvature, a theoretical contact pressure of $2.161\times10^5 psi$ was obtained, compared to a prediction of $2.014\times10^5 psi$ from *Helical3D*.

More details are provided in the Helical3D validation manual [37].

These results lend credence to contact pressure predictions for situations where the load is not distributed uniformly across the face width, where the curvature is not known, or for some other reason simple formulae cannot be used.

5.4 Effect of cutter tip radius on the maximum principal normal stress

We ran the analysis for each of the tip radii for all the mesh templates. The tip radius was varied from $0.01in$ to $0.045in$ in steps of $0.005in$. A quadratic tip modification (Tables 5.8 and 5.9) was applied to the pinion and gear.

The stress values hence obtained are shown in Table 5.10. Figure 5.7 shows a plot of the maximum principal normal stress vs. cutter tip radius. As expected, the stress decreases with

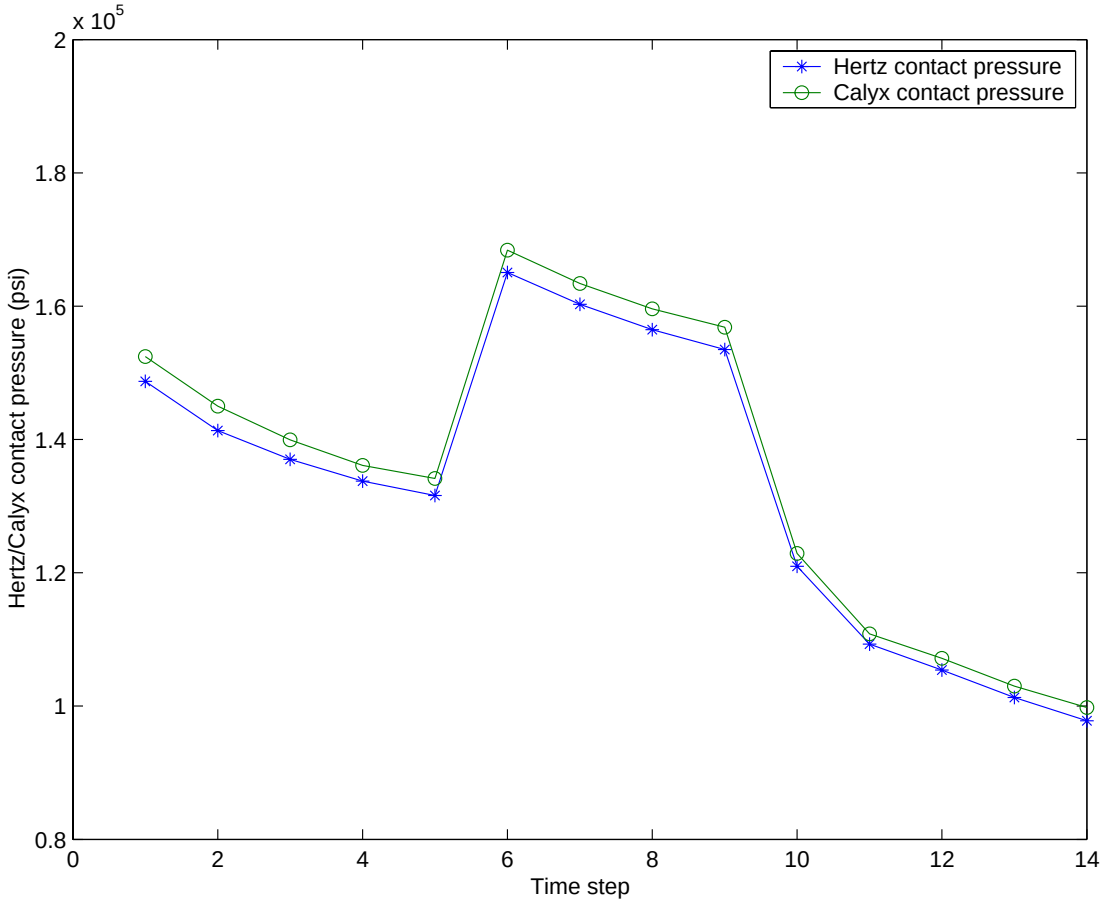

Figure 5.4: A graph comparing *Helical3D*'s and Hertzes contact pressure predictions

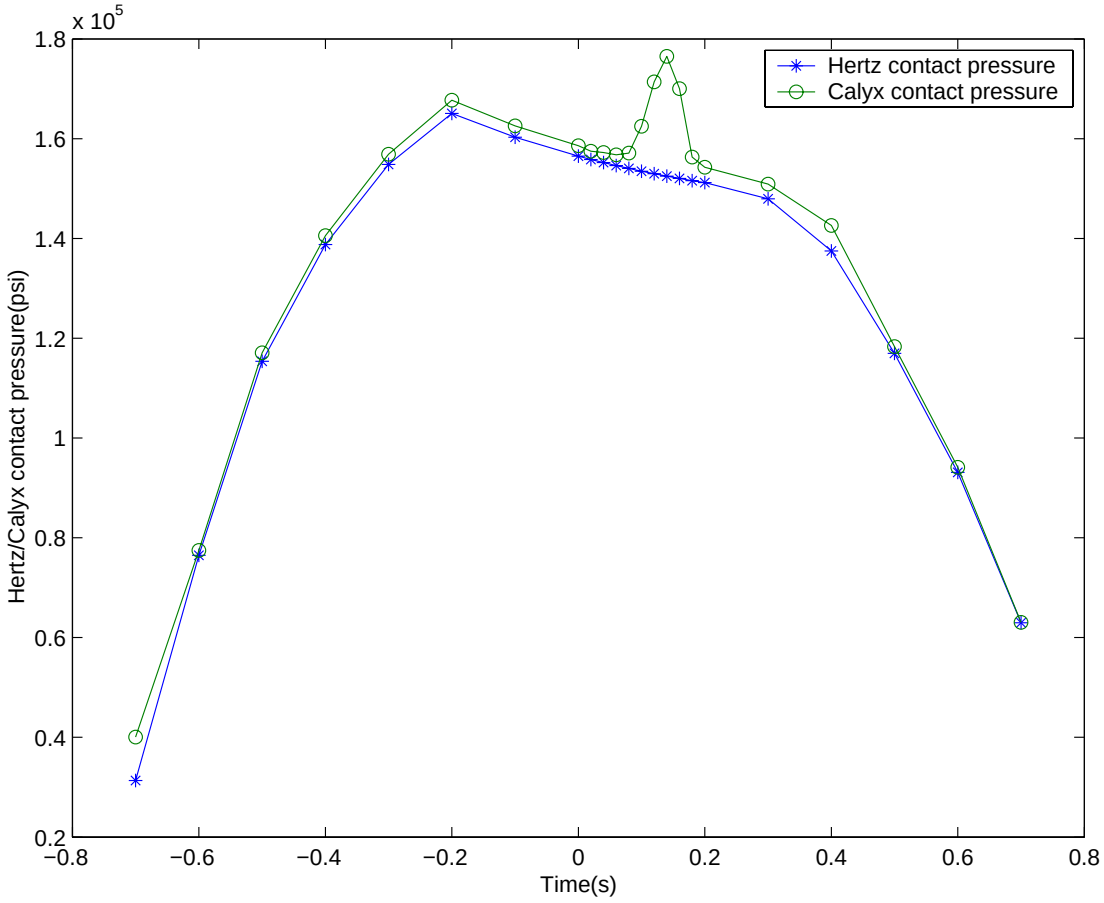

Figure 5.5: A graph comparing *Helical3D*'s and Hertzes contact pressure predictions for linear modification at the teeth

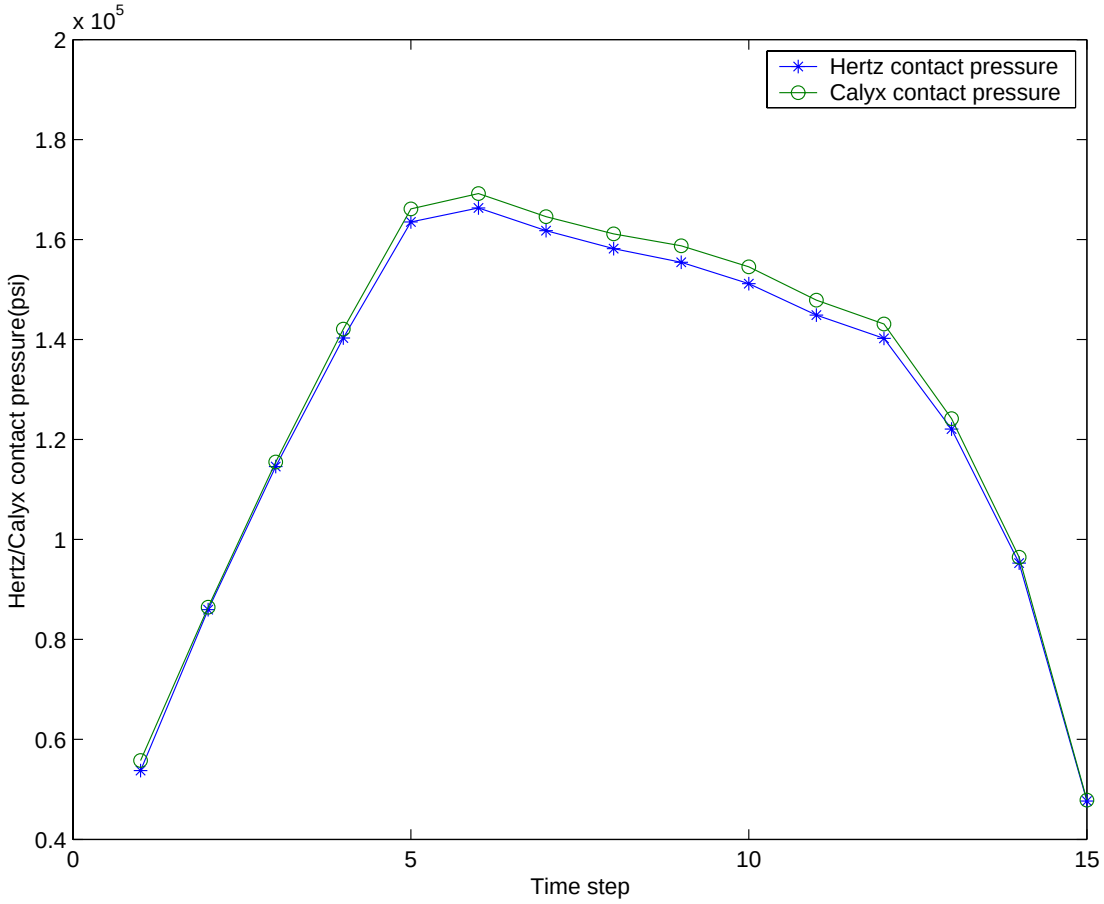

Figure 5.6: A graph comparing *Helical3D*'s and Hertzes contact pressure predictions for quadratic modification at the teeth

Table 5.10: Maximum principal normal stress values for different radii

Tip radius	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
0.010	5.801E+04	5.651E+04	6.099E+04
0.015	5.498E+04	5.397E+04	5.802E+04
0.020	5.230E+04	5.169E+04	5.528E+04
0.025	5.022E+04	4.985E+04	5.303E+04
0.030	4.844E+04	4.816E+04	5.068E+04
0.035	4.691E+04	4.669E+04	4.908E+04
0.040	4.553E+04	4.538E+04	4.731E+04
0.045	4.425E+04	4.417E+04	4.576E+04

increase in the cutter tip radius. From the graph it can be seen that the difference in the results for the medium and fineroot templates is 2.5% at a tip radius of $0.01in$. The difference in the results between fineroot and finest templates is about 7.0% at $0.01in$ tip radius. The agreement is better for larger values of tip radius. For the largest radius ($0.045in$) the difference in the results between medium and fineroot templates is less than 1.0% and that between fineroot and finest templates is about 3.0%.

The time required to run the analysis for each case with the medium, fineroot and finest templates was about $20mins$, $1hr$ and $5hrs$ respectively on a Intel $pentium4$, $1700MHz$ CPU.

5.5 Effect of tooth thickness on the maximum principal normal stress

In order to study the effect of tooth thickness on the stress values we run the analysis for different tooth thicknesses for all the mesh templates. We vary the thickness from $0.1in$ to $0.16in$ in steps of $0.005in$. A quadratic tip modification (Tables 5.8 and 5.9) was applied to the pinion and gear. The tip radius for all the tooth thickness values is $0.02in$. The stress values hence obtained are shown in Table 5.11. Figure 5.8 shows a plot of the maximum principal normal stress against tooth thickness. The stress decreases monotonically with an increase in tooth thickness. From the graph it can be seen that the difference in the results for the medium and fineroot templates is about 1.4% at $0.16in$ tooth thickness. The difference between the fineroot and finest templates is about 6.0%.

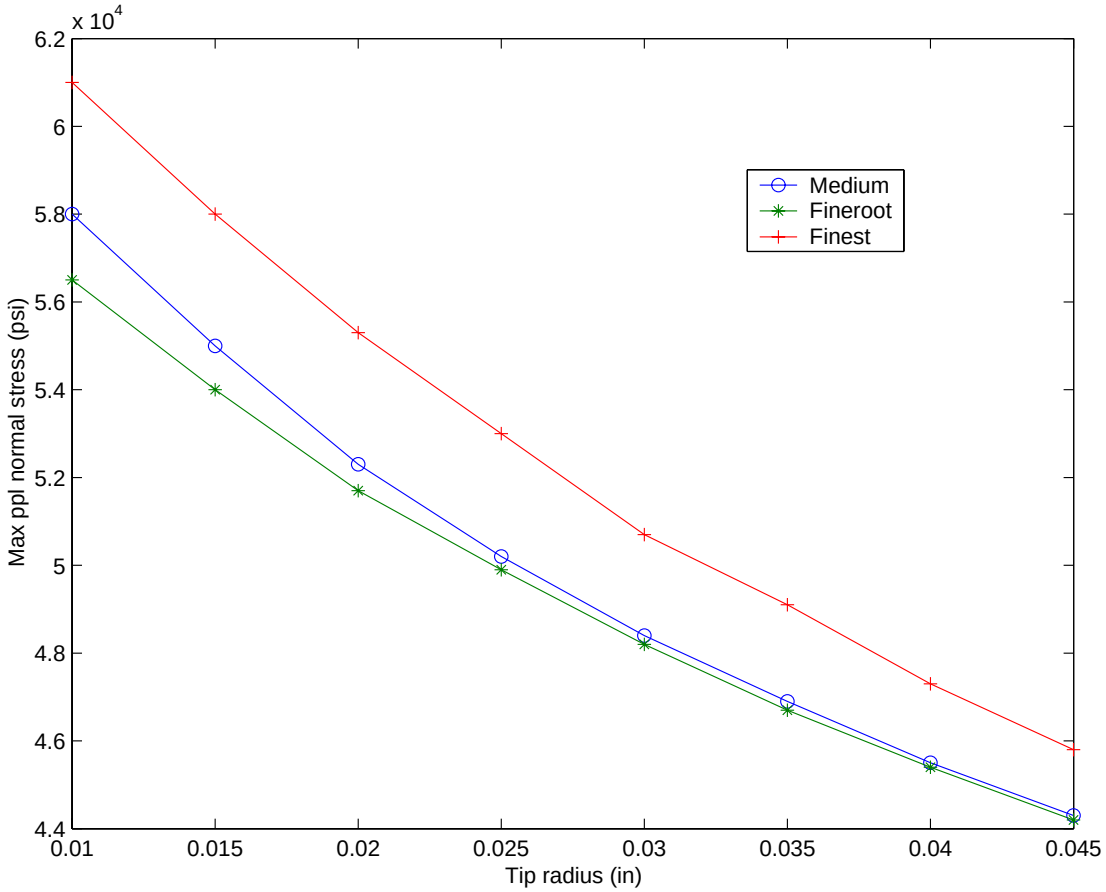

Figure 5.7: Graph of maximum principal normal stresses against tip radii (0.010 in-0.045 in) for medium, fineroot and finest templates

Table 5.11: Maximum principal normal stress values for different tooth thicknesses

Tooth thickness	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
0.100	7.6512E+04	7.7707E+04	7.9807E+04
0.105	7.4098E+04	7.4538E+04	7.6870E+04
0.110	7.1568E+04	7.1727E+04	7.3710E+04
0.115	6.9032E+04	6.8948E+04	7.0545E+04
0.120	6.6528E+04	6.6432E+04	6.9142E+04
0.125	6.4025E+04	6.4041E+04	6.7245E+04
0.130	6.1698E+04	6.1801E+04	6.5013E+04
0.135	5.9655E+04	5.9672E+04	6.2469E+04
0.140	5.7885E+04	5.7643E+04	6.0617.E+04
0.145	5.6152E+04	5.5832E+04	5.9515E+04
0.150	5.4468E+04	5.4080E+04	5.8010E+04
0.155	5.2993E+04	5.2463E+04	5.6264E+04
0.160	5.1600E+04	5.0854E+04	5.4068E+04

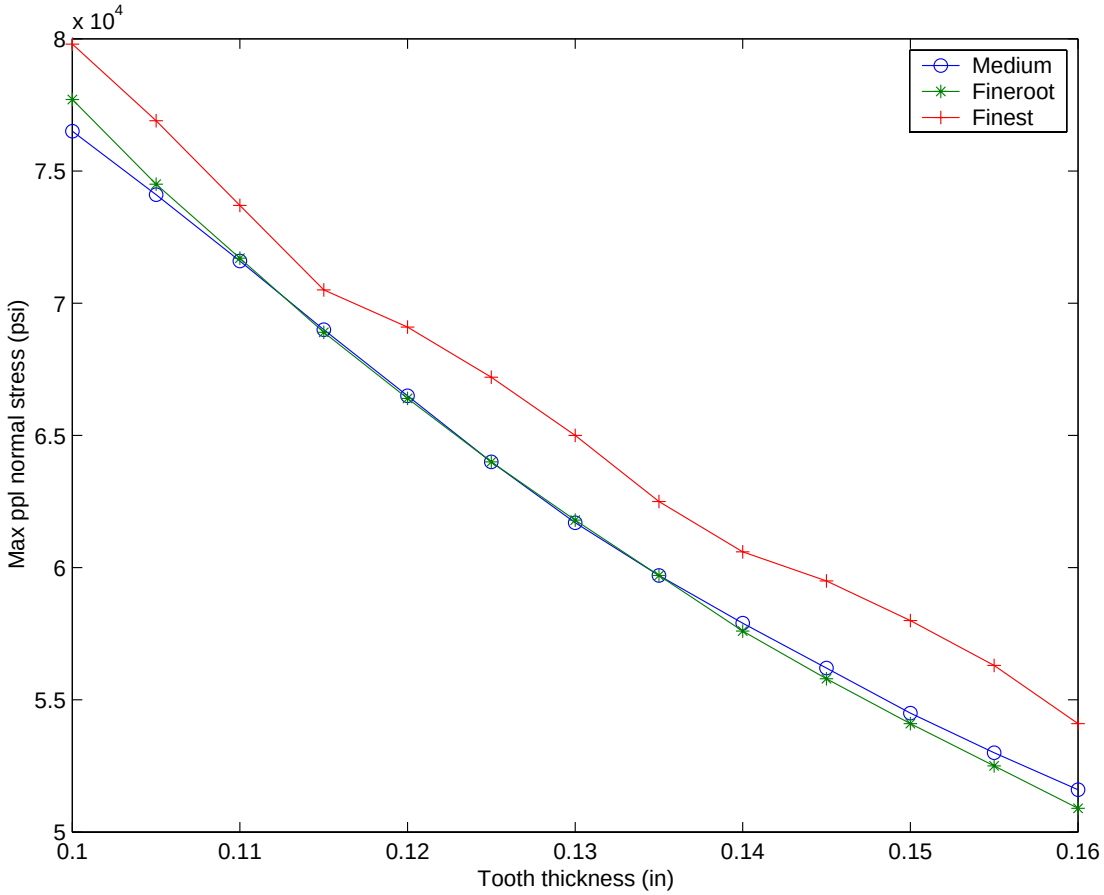

Figure 5.8: Graph of maximum principal normal stresses against tooth thickness (0.10in-0.16in) for medium, fineroot and finest templates

5.6 Effect of number of elements in the face direction on the maximum principal normal stress

In order to study the effect of number of elements along the face width on the stress values we run the analysis with different values for NFACEELEMS, with all the mesh templates. NFACEELEMS is the number of finite elements in the face direction of the tooth. We vary the NFACEELEMS parameter from 2 to 10 in steps of 2. The tip radius and the tooth thickness values for all the test cases are $0.02in$ and $0.15708in$ respectively. The stress values hence obtained are shown in Table 5.12. Figure 5.9 shows a plot of the maximum principal normal stress against the no. of face elements for the medium and fineroot templates. Results for higher elements with the finest mesh could not be obtained due to CPU limitations. From the graph it can be seen that the difference in the results for the medium and fineroot templates is about 1.1% As expected the stress values converge for higher number of elements.

Table 5.12: Maximum principal normal stress values for different number of elements along the face width

Nfaceelems	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
2	5.2390E4	5.1800E4	5.5463E4
4	5.2303E4	5.1691E4	5.5287E4
6	5.2348E4	5.1766E4	-
8	5.2348E4	5.1766E4	-
10	5.2348E4	5.1766E4	-

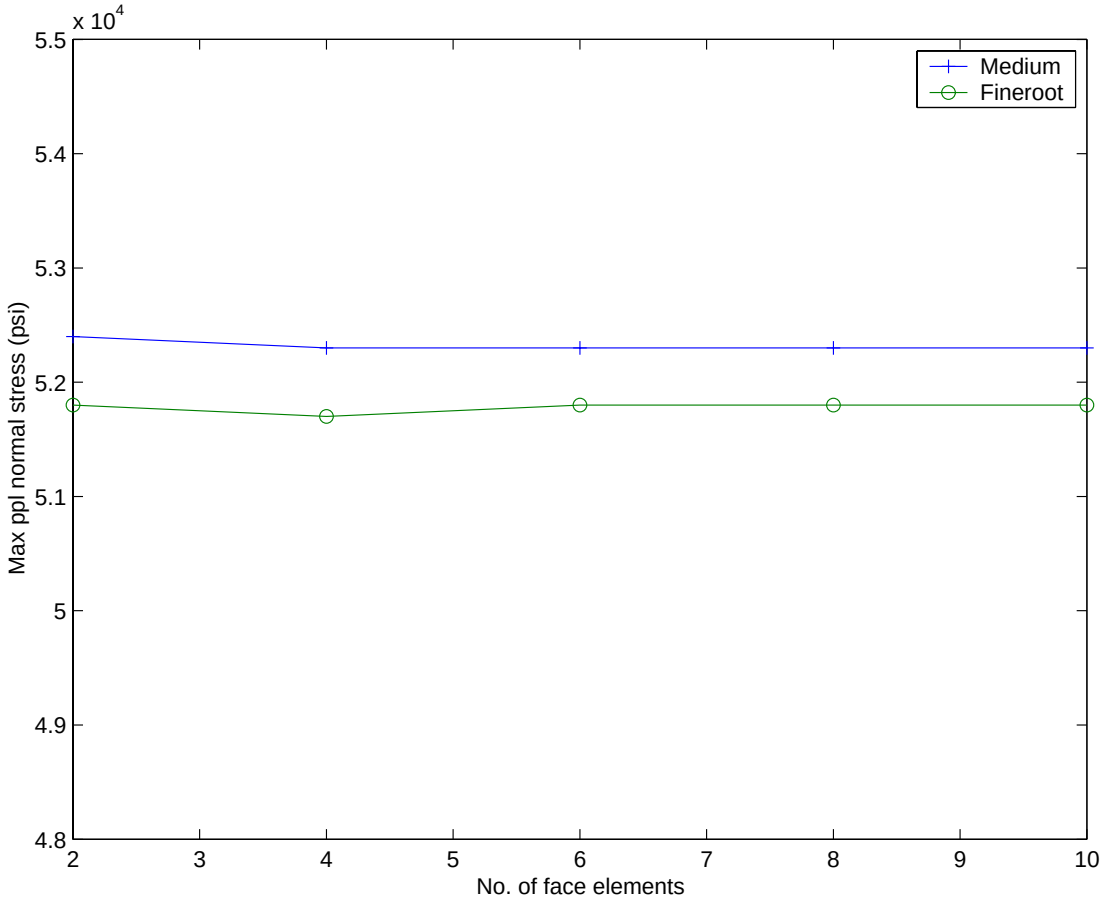

Figure 5.9: Graph of maximum principal normal stresses against No.of elements along the face width for medium and fineroot templates

Table 5.13: Maximum principal normal stress values for different Displ. order

Displ.order	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
3	5.2303E4	5.1691E4	5.5287E4
4	5.2356E4	5.1774E4	-
5	5.2345E4	5.1765E4	-
6	5.2345E4	5.1765E4	-

5.7 Effect of displacement order on the maximum principal normal stress

In order to study the effect of the finite element displacement interpolation order in the face width direction (DISPLORDER), on the stress values we run the analysis for different DISPLORDER for all the mesh templates. With 4 elements along the face width, we vary the DISPLORDER parameter from 3 to 6 in steps of 1. A quadratic tip modification (Tables 5.8 and 5.9) was applied to the pinion and gear. The tip radius and the tooth thickness values for all the test cases are $0.02in$ and $0.15708in$ respectively. The stress values hence obtained are shown in Table 5.13. Figure 5.10 shows a plot of the maximum principal normal stress against the displacement order for the medium and fineroot templates. Results for higher order with the finest mesh could not be obtained due to CPU limitations. From the graph it can be seen that the difference in the results for the medium and fineroot templates is about 1.1%. The results converge for higher order as expected.

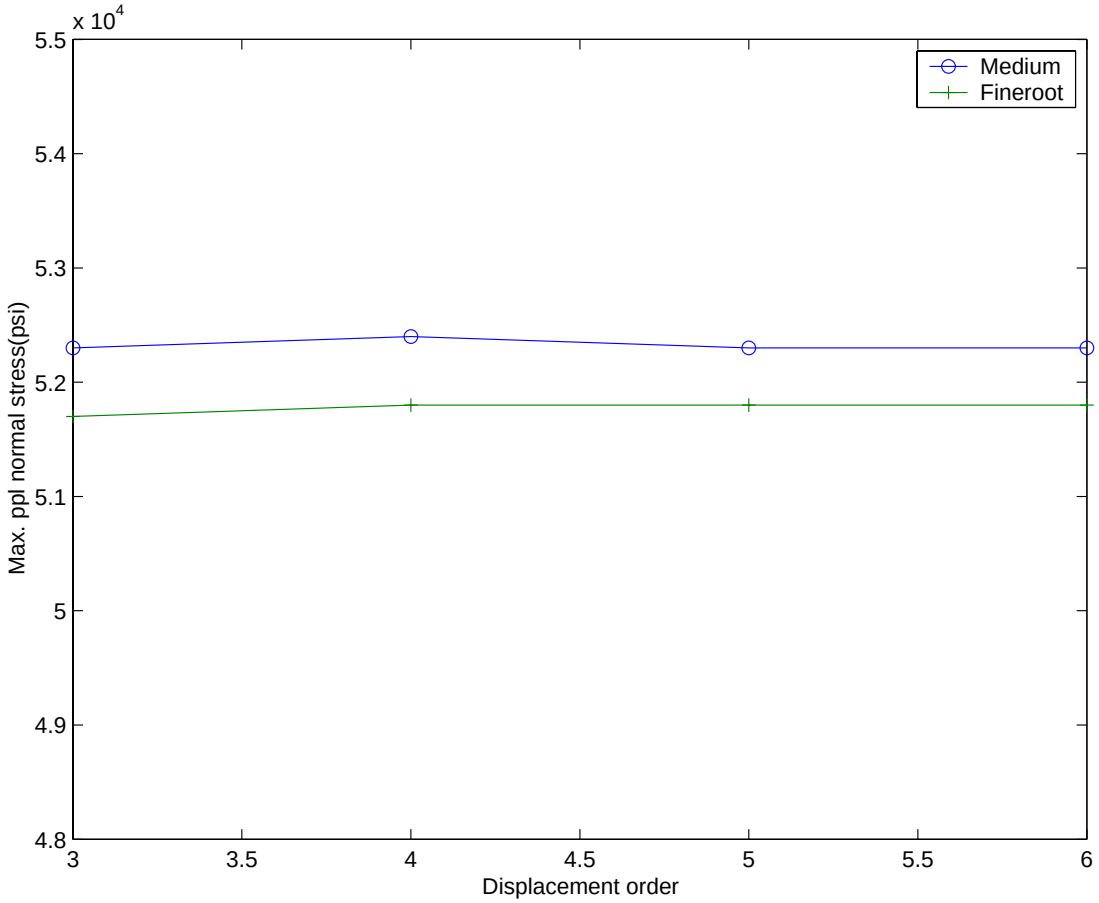

Figure 5.10: Graph of maximum principal normal stresses against displacement order for medium and fineroot templates

Chapter 6

HypoidFaceMilled

HypoidFaceMilled is the software module that we have built for the analysis of spiral-bevel and hypoid pinions and gears cut by the face milling process. Both Formate as well as Generated gear geometries are supported by *HypoidFaceMilled*. The module was built using the standard *Calyx*, *Multyx* and *Guide* sub-components.

The machine kinematics for the face milling process have been described in 2.4. A detailed description of the software package is provided in the *HypoidFaceMilled User's Manual*[38], and detailed validation studies are in the *HypoidFaceMilled Validation Manual*[38]. We will provide a brief summary here.

6.1 Comparison with Experiment

An aerospace spiral bevel gear set is chosen for this validation study because strain gage measurements are available [40] for the pinion stresses. Table 6.1 shows the system level information for this gear set. Table 6.2 shows the pinion data, and Table 6.3 shows the gear data.

The spiral-bevel gear set described above was instrumented by Handschuh [40] with strain gages as shown in Figure 6.1. Figure 6.2 shows the gage measurements at a gear torque of 7840 inch-lbs (Pinion torque 2613.33 inch-lbs).

Figure 6.3 shows the model of the spiral bevel gear set. For the gear, the all nodes on the conical surface at the base of the teeth are constrained. For the pinion, the deformation of the cylindrical surface at the base of the teeth is expressed as a Fourier series in the circular direction and a polynomial series in the axial direction. Those terms of this series expansion that correspond to rigid body motion are constrained, and the remaining terms are left free. Thus the surface is free to deform to any shape, without undergoing rigid body motion. Four finite elements of polynomial order 3 are used along the face width direction of the pinion and

Table 6.1: System Data

Hand of Pinion	Left
Shaft Offset	0
Shaft Angle	90 Deg
Loaded Side of Gear	Convex
Driver	Pinion
Pinion Torque	2613.33 inch-lb (=7840 inch-lb at the Gear)
Coefficient of Friction	0

Table 6.2: Pinion Data

	Concave	Common	Convex
No. of teeth		12	
Transverse Circular Tooth Thickness at Pitch Cone		0.32 inch	
Outer cone distance		3.691 inch	
Face Width		1.0 inch	
Face Angle		22.31667 Deg	
Back Angle		18.433 Deg	
Spiral Angle		35 Deg	
Pitch Angle		18.433 Deg	
Pitch apex beyond Crossing Point		0	
Face apex beyond crossing point		0	
Root apex beyond crossing point		0	
Diameter of cylinder at the base of the tooth		1.138 inch	
Young's Modulus		30.0e6 psi	
Poisson's Ratio		0.3	
	Machine Settings		
Radial setting	2.947802 inch		2.801049 inch
Tilt angle	0		0
Swivel angle	0		0
Blank offset	0.1545759 inch		-0.1742616 inch
Root Angle	16.8666 Deg		16.8666 Deg
Machine Center to Back	-0.04023062 inch		0.05414291 inch
Sliding Base	0.01167273 inch		-0.01570932 inch
Cradle Angle	63.94203 Deg		53.92599 Deg
Ratio of Roll	3.242698536		3.105176807
2C Mod. Roll Coeff.	0		0
6D Mod. Roll Coeff.	0		0
24E Mod. Roll Coeff.	0		0
120F Mod. Roll Coeff.	0		0
H1 Helical motion coeff	0		0
H2 Helical motion coeff	0		0
H3 Helical motion coeff	0		0
V1 Vert. motion coeff	0		0
V2 Vert. motion coeff	0		0
V3 Vert. motion coeff	0		0
	Cutter Geometry		
Cutter Type	Straight		Straight
Point Radius	2.965621 inch		3.071306 inch
Blade Angle	18.04567 Deg		24.33742 Deg
Edge Radius	0.045 inch		0.045 inch

Table 6.3: Gear Data

	Concave	Common	Convex
No. of teeth		36	
Transverse Circular Tooth Thickness at Pitch Cone		0.15 inch	
Outer cone distance		3.691 inch	
Face Width		1.0 inch	
Face Angle		72.5 Deg	
Back Angle		71.5666 Deg	
Spiral Angle		35 Deg	
Pitch Angle		71.5666 Deg	
Pitch apex beyond Crossing Point		0	
Face apex beyond crossing point		0	
Root apex beyond crossing point		0	
Angle of Cone at the base of the tooth		61.5 Deg	
Young's Modulus		30.0e6 psi	
Poisson's Ratio		0.3	
	Machine Settings		
Type of surface (Generated/Formate)	Generated		Generated
Radial setting	2.85995 inch		2.85995 inch
Tilt angle	0		0
Swivel angle	0		0
Blank offset	0		0
Root Angle	67.68333 Deg		67.68333 Deg
Machine Center to Back	0		0
Sliding Base	0		0
Cradle Angle	59.2342023 Deg		59.2342023 Deg
Ratio of Roll	1.051674445		1.051674445
2C Mod. Roll Coeff.	0		0
6D Mod. Roll Coeff.	0		0
24E Mod. Roll Coeff.	0		0
120F Mod. Roll Coeff.	0		0
H1 Helical motion coeff	0		0
H2 Helical motion coeff	0		0
H3 Helical motion coeff	0		0
V1 Vert. motion coeff	0		0
V2 Vert. motion coeff	0		0
V3 Vert. motion coeff	0		0
	Cutter Geometry		
Cutter Type	Straight		Straight
Point Radius	3.0325 inch		2.9675 inch
Blade Angle	22.0 Deg		22.0 Deg
Edge Radius	0.001 inch		0.001 inch

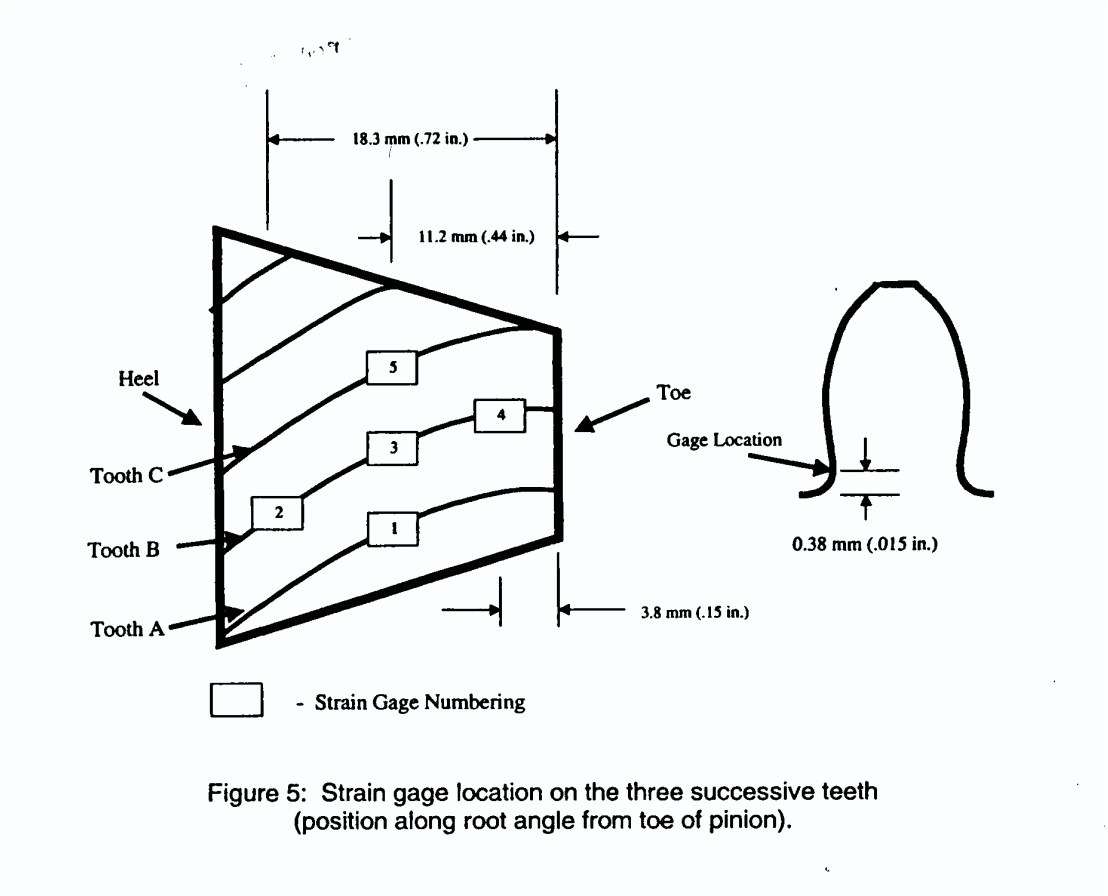

Figure 5: Strain gage location on the three successive teeth (position along root angle from toe of pinion).

Figure 6.1: Strain Gage Location in Experiment. (Reproduced from Handschuh [40])

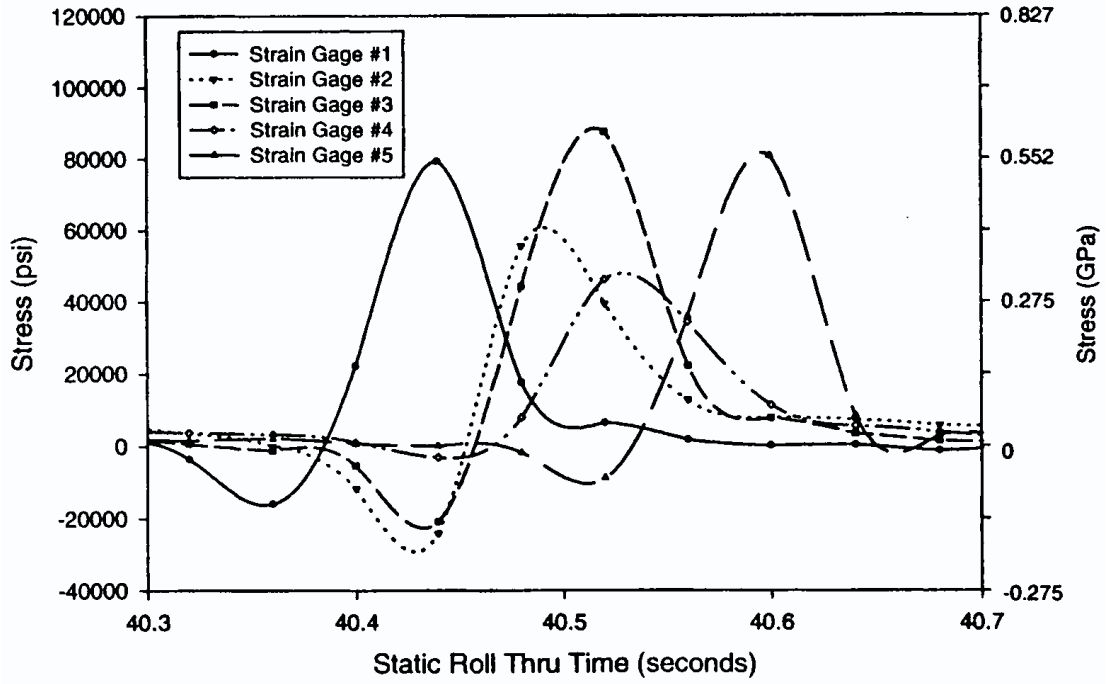

Figure 6.2: Strain gage readings at 7840 inch-lb gear torque. (Reproduced from Handschuh [40])

gear teeth.

The contact pattern obtained after analysis is shown in Figure 6.4. Figure 6.5 and Figure 6.6 show the variation of the maximum principal normal stress along the surface of this model for one position on the gear and pinion, respectively. The load distribution is also shown. Figure 6.7 shows the tooth load as a function of time for one tooth cycle. Figure 6.8 shows the stress in the predicted strain gage readings at various points along the profile at the mid-face cross section. (Gages 1,3 and 5). Figure 6.9 shows similar data at Zeta=0.44 (Gage 2) and Figure 6.10 shows predictions at Zeta=-0.701 (Gage 4). The strain gage measurements in Handschuh [40] lie roughly halfway between the lowest and highest stresses along the profile direction, and the shapes of the predictions are very similar to the measurements.

6.2 Effect of varying the cutter tip radius on the maximum principal normal stress

In order to study the effect of cutter tip radius on the stress values we run the analysis for different tip radii for all the mesh templates. We vary the tip radius from $0.005in$ to $0.085in$ in steps of $0.005in$. The stress values obtained are shown in Table 6.4. Figure 6.11 shows a plot of Tip radius against the maximum principal normal stress. It can be concluded from the graph that as you go on increasing the tip radius, the stress decreases. From the graph it can be seen that the difference in the results for the medium and fineroot templates is 9.0% at a tip radius of $0.005in$. The difference in the results between fineroot and finest templates is about 4.0% at $0.005in$ tip radius. As expected, the agreement is better for larger values of tip radius.

Table 6.5 shows the results for the stresses over a much smaller range. The tip radius is varied from $0.04in$ to $0.045in$ in steps of $0.0005in$. Even for such a small change in tip radius,

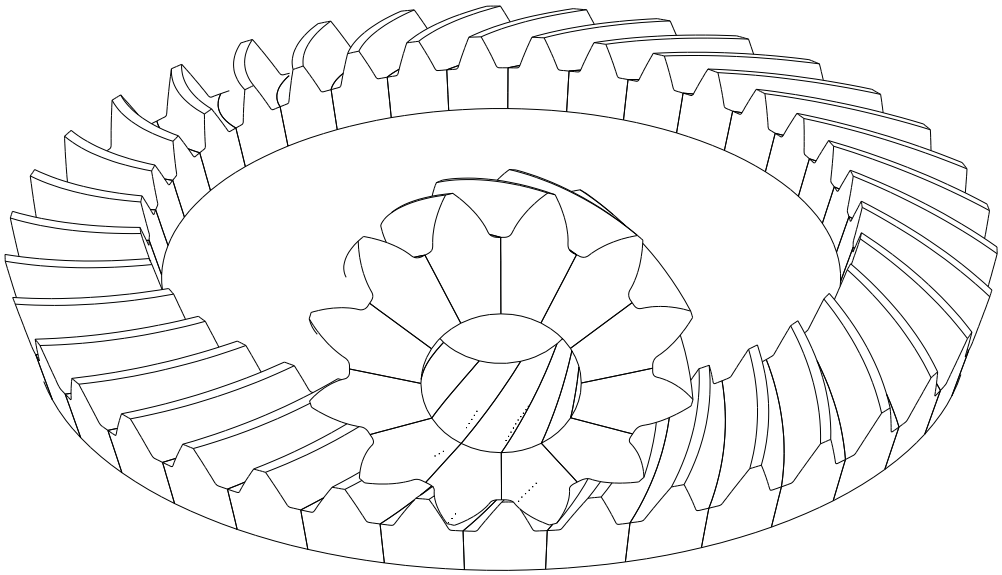

Figure 6.3: The finite element model created by *HypoidFaceMilled*.

the change in stresses is monotonously decreasing.

The time required to run the analysis for each case with the medium, fineroot and finest templates was about $20mins$, $1hr$ and $4hrs$ respectively on an Intel $Pentium4$, $1700MHz$ *cpu*.

6.3 Effect of varying the tooth thickness on the maximum principal normal stress

In order to study the effect of tooth thickness on the stress values we run the analysis for different tooth thicknesses for all the mesh templates. We vary the thickness from $0.32in$ to $0.28in$ in steps of $0.005in$. The tip radius for all the tooth thickness values is $0.045in$. The stress values hence obtained are shown in Table 6.6. Figure 6.12 shows a plot of the maximum principal normal stress against tooth thickness. It can be concluded from the graph that as you go on decreasing the thickness, the stress increases. From the graph it can be seen that the difference in the results for the medium and fineroot templates is about 4.0%. The difference between the fineroot and finest templates is about 1.0%.

6.4 Effect of varying the number of elements in the face direction on the maximum principal normal stress

In order to study the effect of number of elements along the face width on the stress values we run the analysis for different NFACEELEMS for all the mesh templates. We vary the NFACEELEMS parameter from 2 to 10 in steps of 2. The tip radius and the tooth thickness values for all the test cases are $0.045in$ and $0.32in$ respectively. The stress values hence obtained are shown in Table 6.7. Figure 6.13 shows a plot of the maximum principal normal stress against the no. of

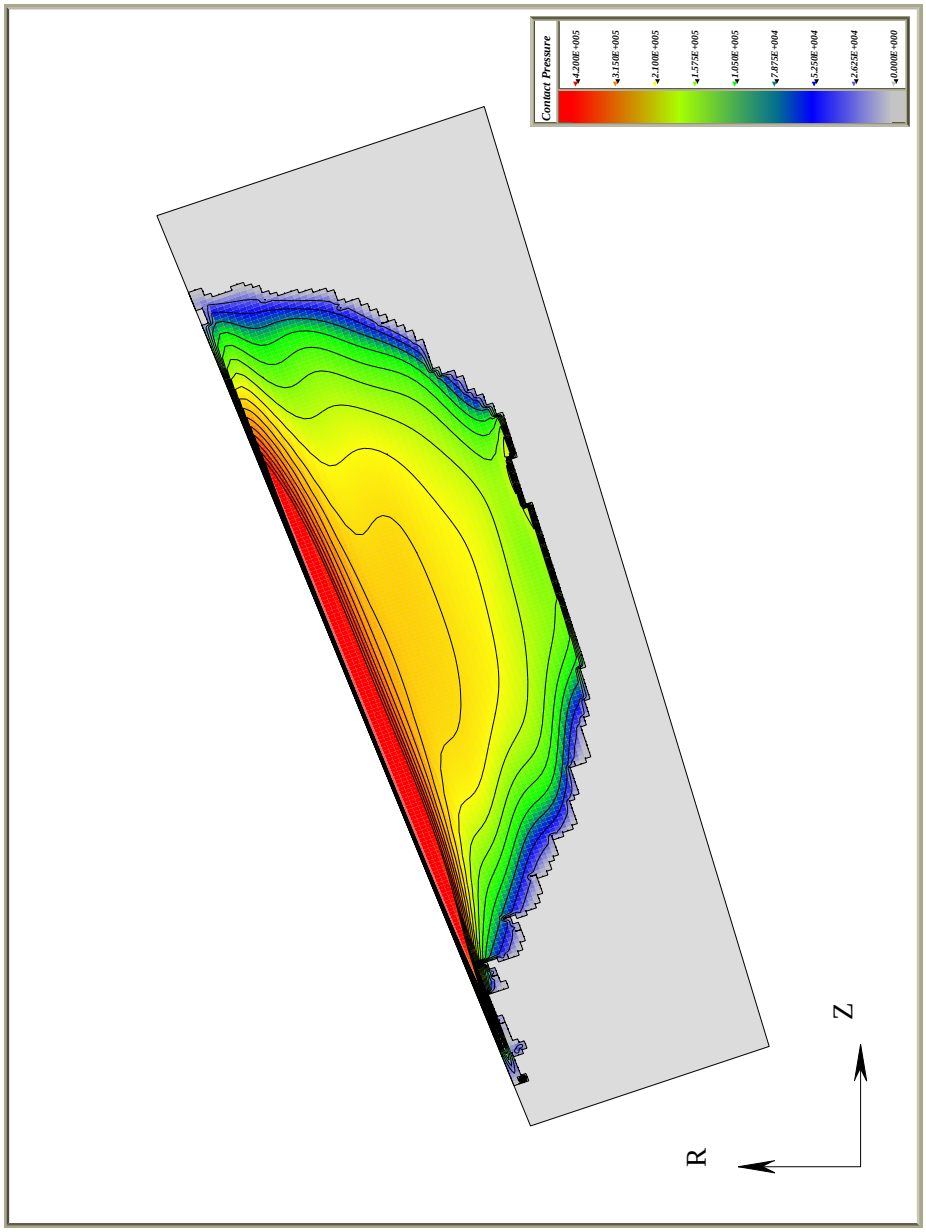

Figure 6.4: The contact pattern.

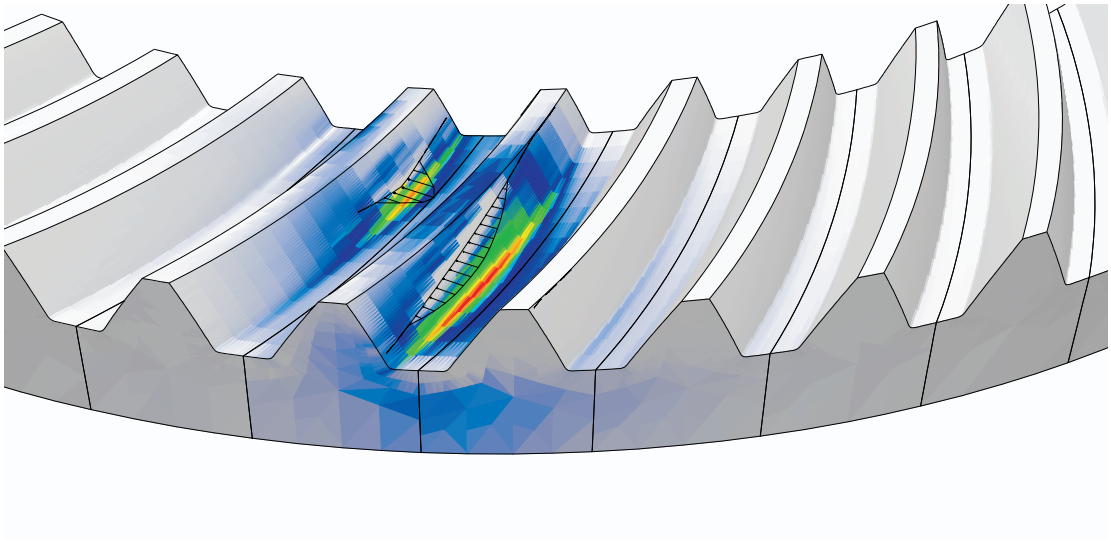

Figure 6.5: Distribution of the maximum principal normal stress along the surface of the gear. The load distribution is also shown.

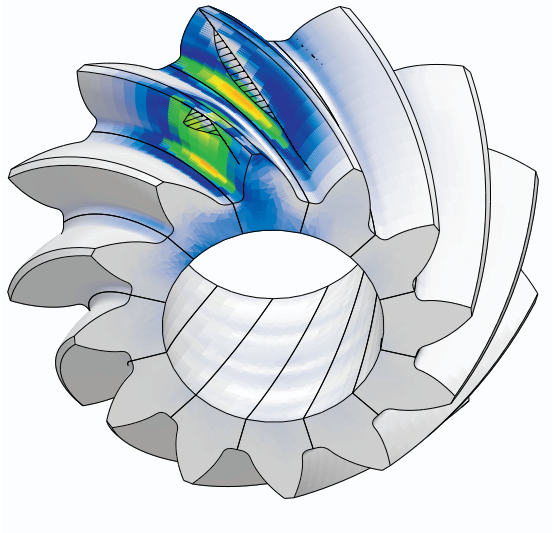

Figure 6.6: Distribution of the maximum principal normal stress along the surface of the pinion. The loads acting on the pinion are also shown.

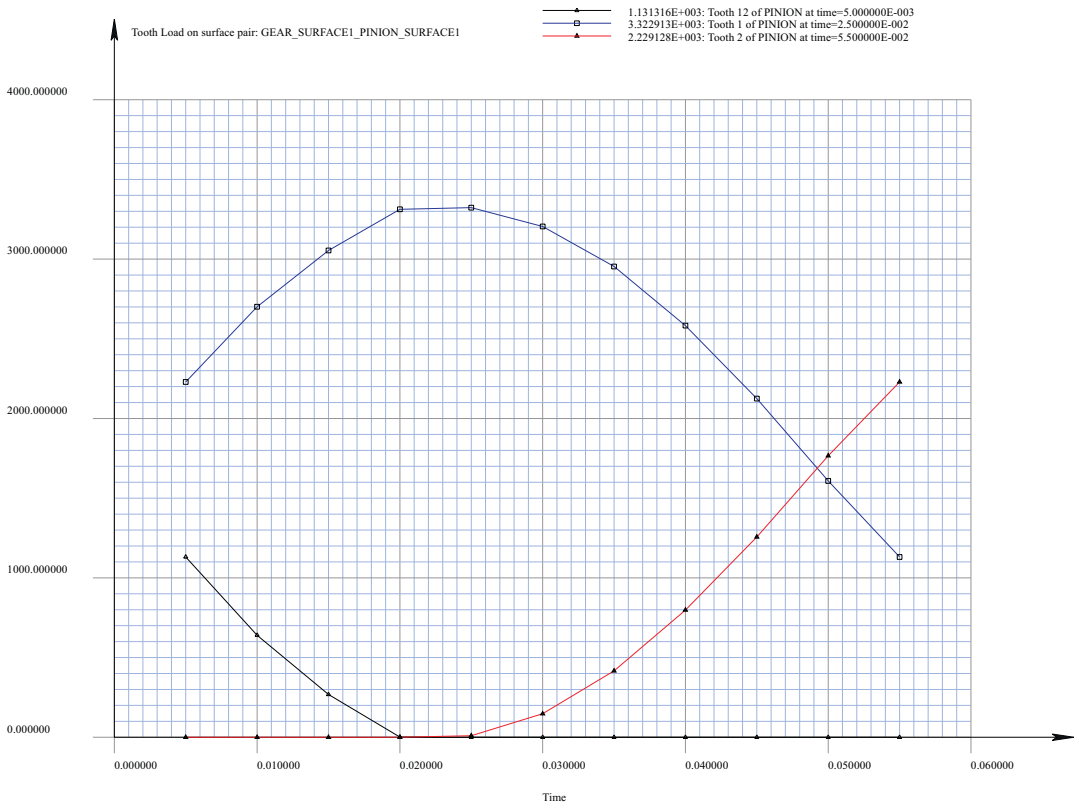

Figure 6.7: Predicted tooth load as a function of time, shown over one tooth cycle.

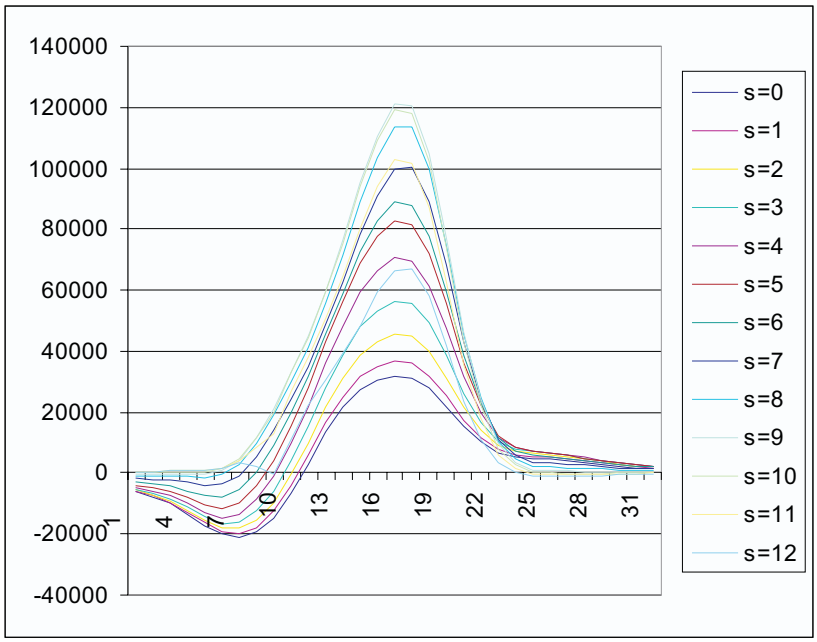

Figure 6.8: Stress predictions for the mid-face (tooth cross-section: Zeta=0.0) Max stress is 119,000 psi at s=10.0.

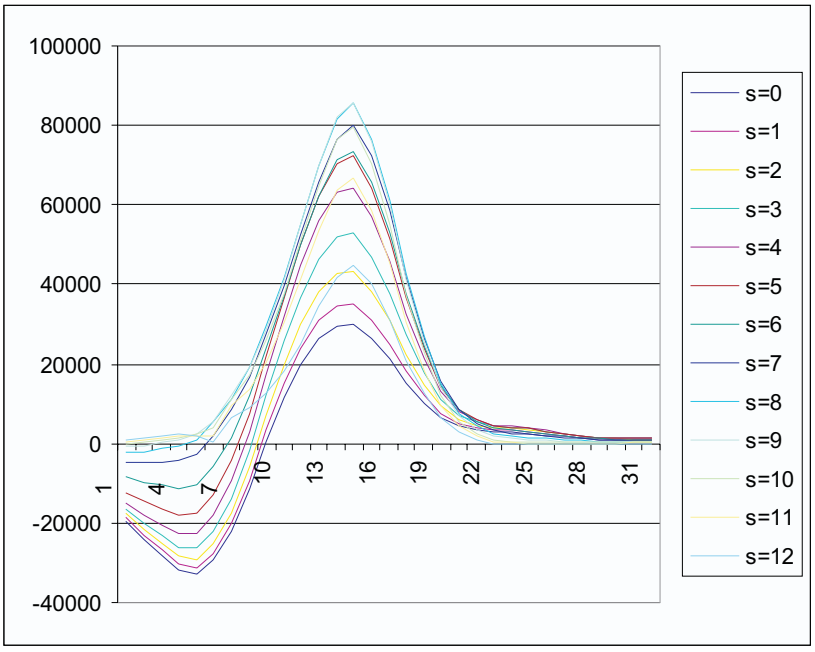

Figure 6.9: Stress predictions by *HypoidFaceMilled* near the heel end, (tooth cross-section: Zeta=0.440). Max stress is 86,000 psi at s=9.0.

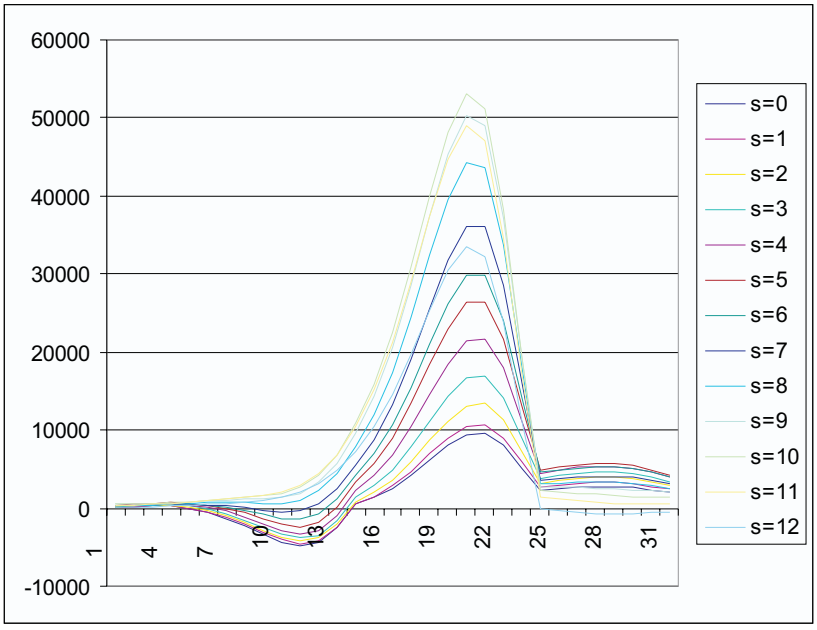

Figure 6.10: Stress predictions by *HypoidFaceMilled* near the toe end. (tooth cross-section: Zeta=-0.701). Max stress is 53,000 psi at s=10.0.

Table 6.4: Maximum principal normal stress values for different radii

Tip radius	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
0.005	1.711E+05	1.892E+05	1.974E+05
0.01	1.638E+05	1.760E+05	1.812E+05
0.015	1.567E+05	1.661E+05	1.683E+05
0.02	1.500E+05	1.565E+05	1.581E+05
0.025	1.434E+05	1.482E+05	1.495E+05
0.03	1.370E+05	1.412E+05	1.421E+05
0.035	1.311E+05	1.351E+05	1.356E+05
0.04	1.259E+05	1.296E+05	1.299E+05
0.045	1.215E+05	1.266E+05	1.248E+05
0.05	1.192E+05	1.263E+05	1.224E+05
0.055	1.190E+05	1.260E+05	1.222E+05
0.06	1.187E+05	1.257E+05	1.220E+05
0.065	1.184E+05	1.254E+05	1.217E+05
0.07	1.174E+05	1.240E+05	1.205E+05
0.075	1.142E+05	1.208E+05	1.175E+05
0.08	1.099E+05	1.165E+05	1.133E+05
0.085	1.039E+05	1.104E+05	1.075E+05

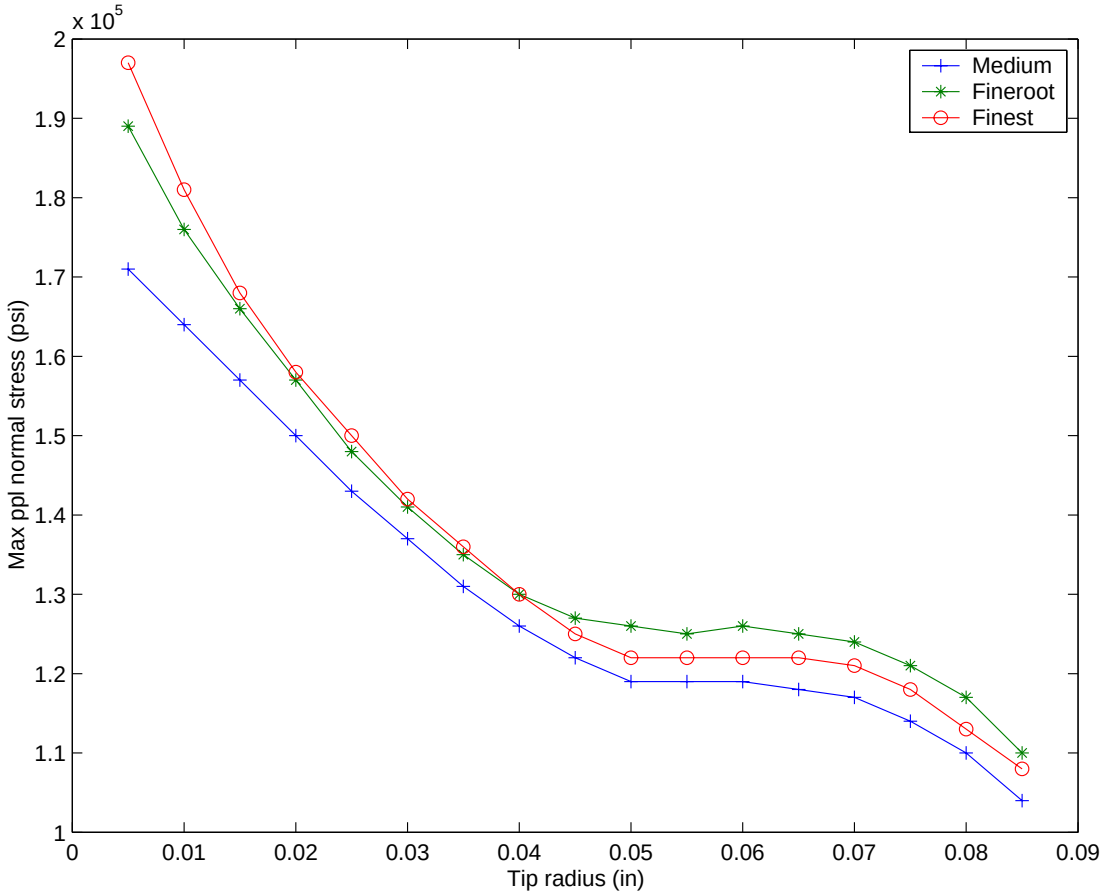

Figure 6.11: Graph of maximum principal normal stress against tip radii ($0.005in - 0.085in$) for medium, fineroot and finest templates

Table 6.5: Maximum principal normal stress values for different radii

Tip radius	Stress with medium.tpl
0.04	1.259E+05
0.0405	1.255E+05
0.041	1.250E+05
0.0415	1.245E+05
0.042	1.241E+05
0.0425	1.236E+05
0.043	1.232E+05
0.0435	1.228E+05
0.044	1.223E+05
0.0445	1.219E+05
0.045	1.215E+05

Table 6.6: Maximum principal normal stress values for different tooth thicknesses

Tooth thickness	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
0.32	1.215E5	1.266E5	1.248E5
0.315	1.231E5	1.263E5	1.261E5
0.31	1.248E5	1.290E5	1.270E5
0.305	1.273E5	1.313E5	1.289E5
0.3	1.294E5	1.330E5	1.312E5
0.295	1.313E5	1.349E5	1.340E5
0.29	1.342E5	1.375E5	1.364E5
0.285	1.366E5	1.396E5	1.381E5
0.28	1.385E5	1.412E5	1.394E5

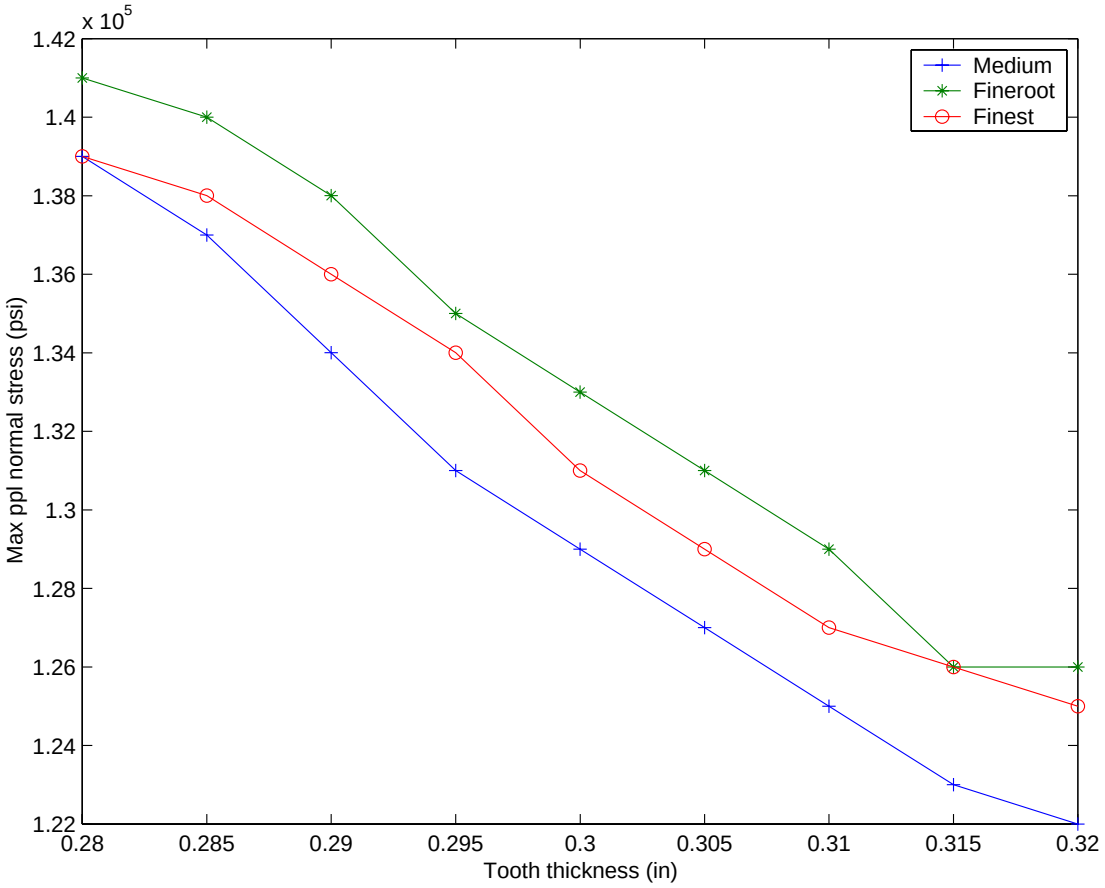

Figure 6.12: Graph of maximum principal normal stress against tooth thickness (0.32in-0.28in) for medium, fineroot and finest templates

Table 6.7: Maximum principal normal stress values for different number of elements along the face width

Nfaceelems	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
2	1.207E5	1.239E5	1.242E5
4	1.215E5	1.266E5	1.248E5
6	1.212E5	1.243E5	-
8	1.211E5	1.243E5	-
10	1.211E5	1.242E5	-

Table 6.8: Maximum principal normal stress values for different displacement order values

Displ.order	Stress with medium.tpl	Stress with fineroot.tpl	Stress with finest.tpl
3	1.215E5	1.266E5	1.248E5
4	1.211E5	1.243E5	-
5	1.211E5	1.242E5	-
6	1.211E5	1.242E5	-

face elements for the medium and fineroot templates. Results for higher elements with the finest mesh could not be obtained due to CPU limitations. From the graph it can be seen that the difference in the results for the medium and fineroot templates is about 2.5%. As expected the stress values converge for higher number of elements.

6.5 Effect of varying the displacement order on the maximum principal normal stress

In order to study the effect of the displacement order on the stress values we run the analysis for different DISPLORDER for all the mesh templates. With 4 elements along the face width, we vary the DISPLORDER parameter from 3 to 6 in steps of 1. The tip radius and the tooth thickness values for all the test cases are $0.045in$ and $0.32in$ respectively. The stress values hence obtained are shown in Table 6.8. Figure 6.14 shows a plot of the maximum principal normal stress against the displ. order for the medium and fineroot templates. Results for higher order with the finest mesh could not be obtained due to CPU limitations. From the graph it can be seen that the difference in the results for the medium and fineroot templates is about 4.0%. The results converge for higher order as expected.

6.6 Conclusions

The stress values have been shown to converge with increasing refinement of the finite element mesh. For typical tip radius values, this convergence study shows that we have a discretization error of about 4.0% in the coarsest mesh (medium.tpl) and less than 1.0% in the intermediate mesh (fineroot.tpl). We feel that the error in the finest mesh (finest.tpl) is much less than 1.0%. The stress values converge with higher elements along the face direction and also with a higher order Fourier series. The error in the results for less number of face elements and less displ.

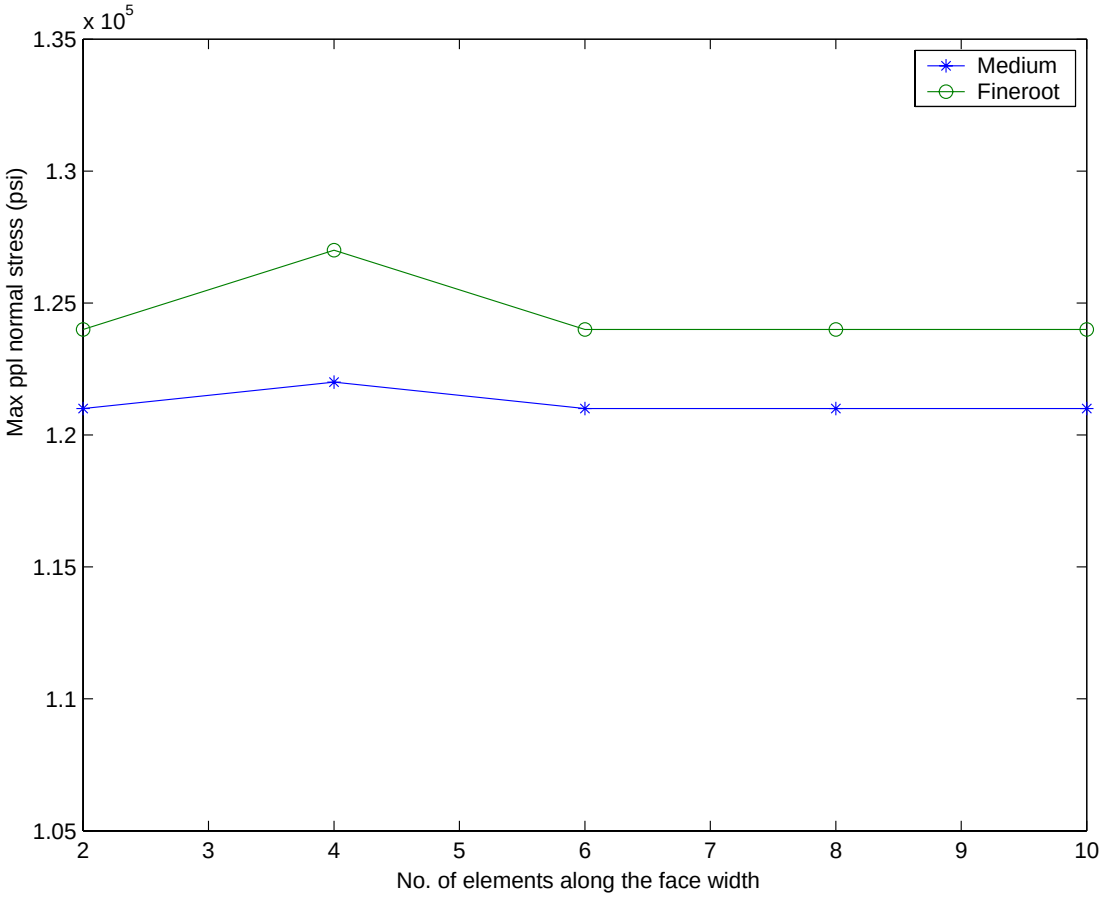

Figure 6.13: Graph of maximum principal normal stress against number of elements along the face width for medium and fineroot templates

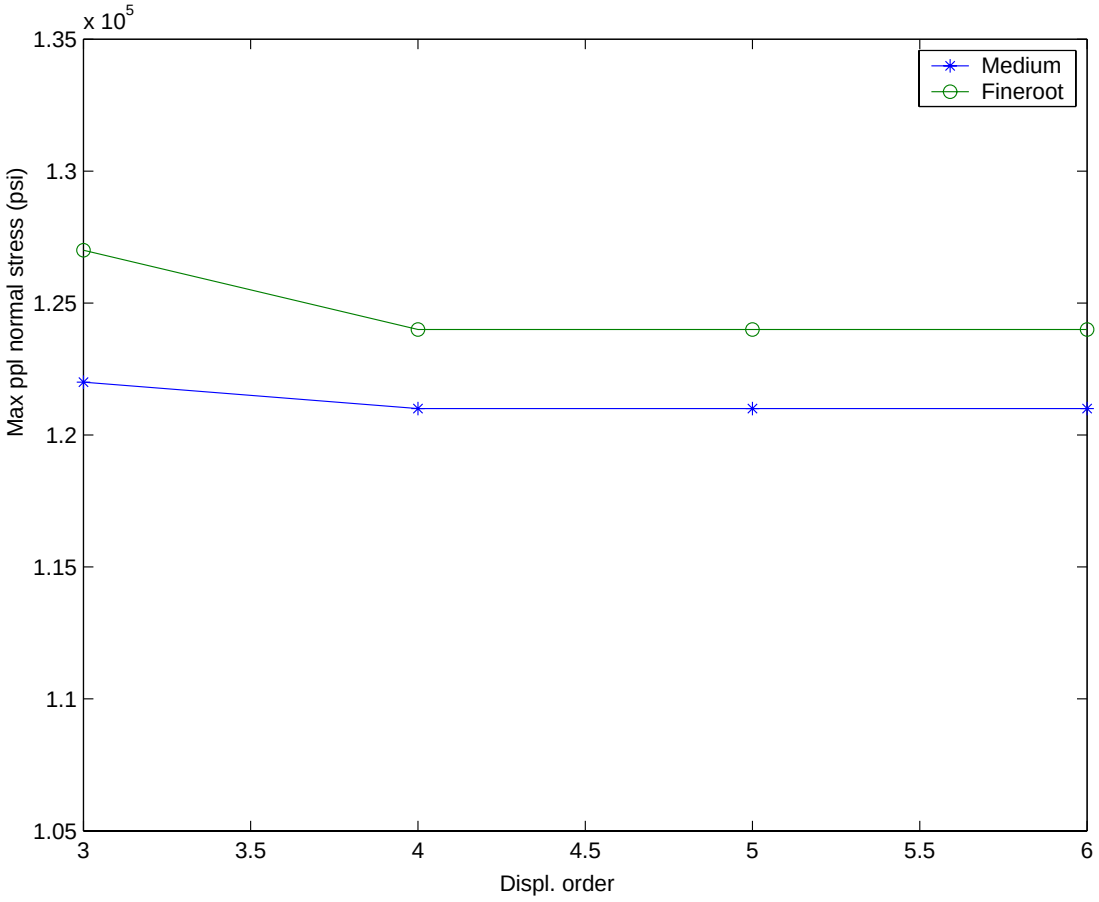

Figure 6.14: Graph of maximum principal normal stress against displacement order for medium and fineroot templates

order is about 4.0% for the medium template and less than 1.0% for the fineroot template.

Chapter 7

Planetary3D

The *Planetary3D* software package, as its name implies, was built to allow us to analyze three-dimensional models of planetary gear systems.

We have implemented detailed rolling element bearing models in this package. Unlike the *Planetary2D* package, detailed finite element models of housings and carriers can be included in *Planetary3D*. Also, we have made it general enough that it is not limited to simple planetary systems. Any of the planetary sets shown in Figure 7.1 can be built. The only limitation that *Planetary3D* places is that system be purely a parallel axis system. This limitation is removed in the *Transmission3D* package.

Planetary3D is implemented using the same basic software platform used by all our other computer packages. This platform consists of the multi-body analysis program *Calyx* which forms the analysis 'engine'. *Multyx* reads the system definition file for *Planetary3D* and exposes the package specific features to the user. *Guide* provides the user-friendly graphical interface.

We provide a brief overview of *Planetary3D* here. Details can be found in the *Planetary3D User's Manual*[43].

The *Planetary3D* model consists of housings, rotors, and connectors. The housing is a finite element model that provides a place for the rtors to connect. The rotors comprise of shafts, carriers with planetary pinions, sun gears and ring gears.

7.1 Countershaft System

Figure 7.2 shows a very simple model built in *Planetary3D*. It has no housing and three rotors. The input and output rotors each have one 'sun' sitting on one shaft. An intermediate rotor has one shaft with two 'sun' gears. Each shaft is supported by two connectors. The connectors are modeled as stiffness matrices provided by the user. These stiffness matrices model the behavior of bearings.

Detailed bearing models can be used in place of any or all of the stiffness type connectors. Figure 7.3 shows a rotor that is supported by a tapered roller bearing model. Figure 7.4 shows the load distribution on the individual rollers of the bearing.

7.2 Split Path System

Systems with multiple power paths are allowed. The system shown in Figure 7.5 is a split-path system with two intermediate rotors. The power flows from the input rotor to the output through both.

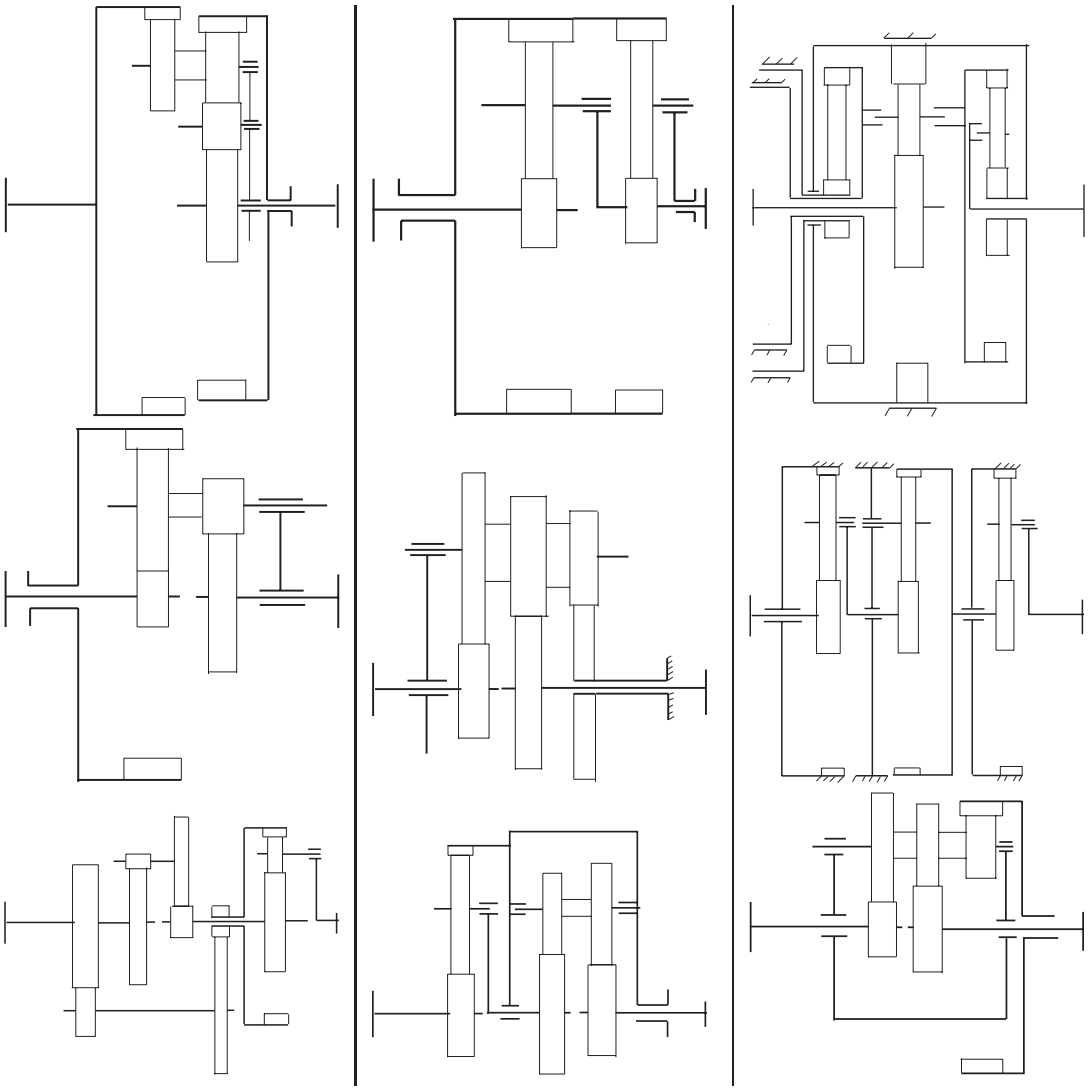

Figure 7.1: Examples of planetary systems that can be modeled using the *Planetary3D package*

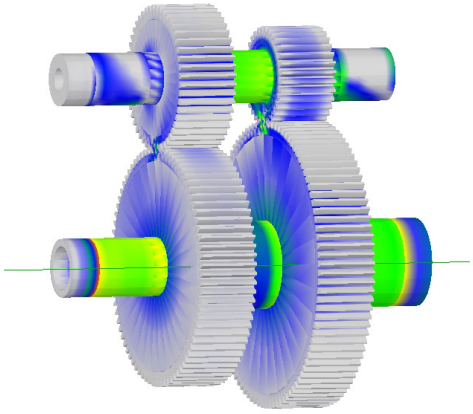

Figure 7.2: A countershaft system model.

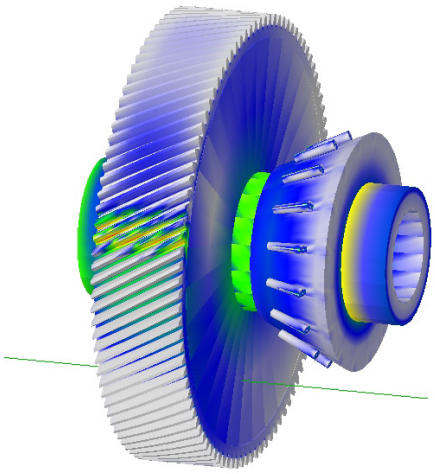

Figure 7.3: A rotor with a tapered roller bearing.

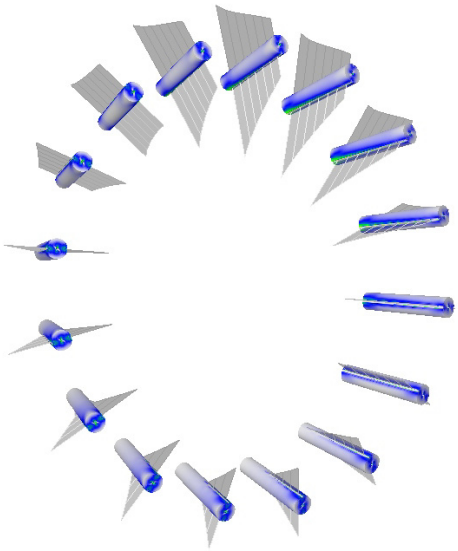

Figure 7.4: Load distribution on the rollers with the axial load carried only by the tapered bearing

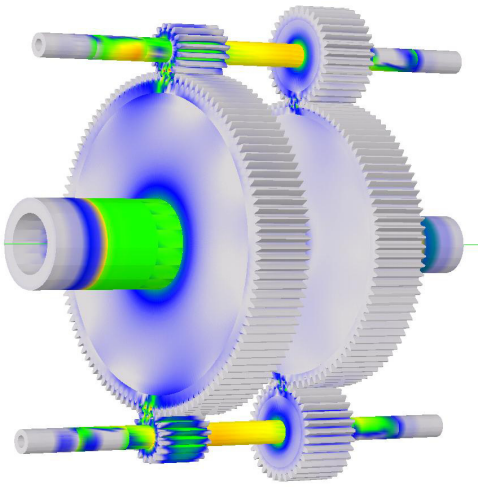

Figure 7.5: Contour showing the maximum principal normal stress contour in the split path gear system

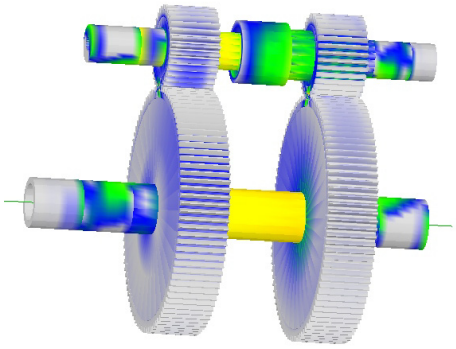

Figure 7.6: The maximum principal normal stresses in the recirculating power gear model for case1

7.3 Power Recirculating System

By specifying a non-zero torsional stiffness for a connector, it can also be used as a coupling. Figure 7.6 shows a system in which a connector with a torsional stiffness and a torsional pre-load has been used to model a recirculating power rig.

7.4 Simple Planetary System

The simplest possible planetary system model that can be built is shown in Figure 7.7. In this system, the outer diameter of the ring gear is constrained. There is one carrier that is rigid, and therefore needs no finite element model. There are four pinions on the carrier. The connection between the pinions and the the carrier is modeled by a stiffness matrix.

Instead of stiffness matrices, detailed bearing models can be used to connect the pinions to the carrier, as shown in Figures 7.8 and 7.9. Figure 7.10 shows the load distribution across the bearing needles under the pinion.

Another variation of a planetary system is depicted in Figure 7.11. The planetary carrier is rigid. The pinons have two decks instead of one. There is no ring gear, and there are two sun gears. The carrier serves as the power input, and the power flows out through the two sun gears. Such an arrangement is commonly used in planetary differential gear sets.

7.5 Detailed Automotive Transfer Case Model

In systems with flexible carriers and housings, finite element models of the carriers and housings must be introduced. We carried out a study of such a system in order to exercise the *Planetary3D* package. We obtained *NASTRAN* models of the housing and carrier of a real life automotive transfer case, converted these models to the *Calyx* format, and loaded them into a *Planetary3D* model. The shafting and all the gears were created by *Planetary3D*. This transfer case model is shown in Figures 7.12 through 7.16.

The transfer case contains a simple planetary system with one sun, one ring gear, and one carrier with four pinions. The outer diameter of the ring gear was made flexible, and was supported in the housing through splines, as shown in Figure 7.13.

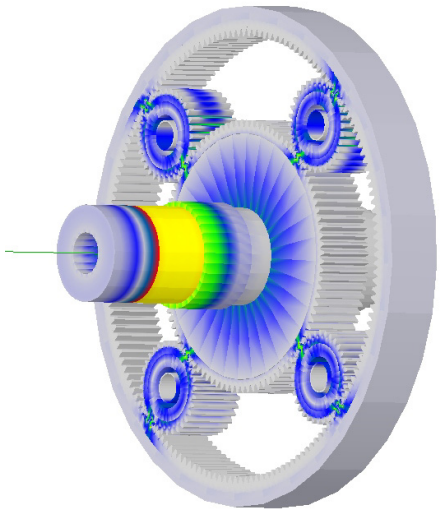

Figure 7.7: A simple planetary reduction gear set

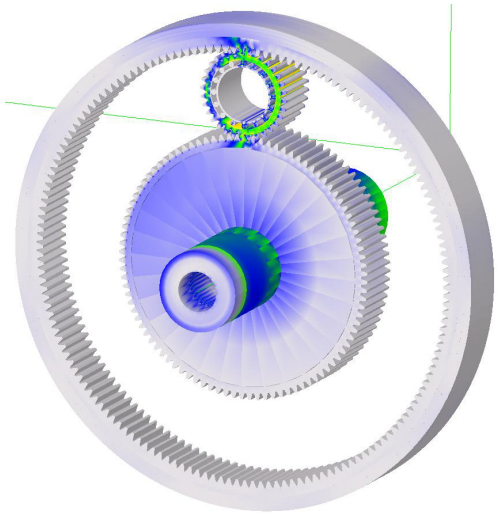

Figure 7.8: A planetary pinion with a detailed bearing model

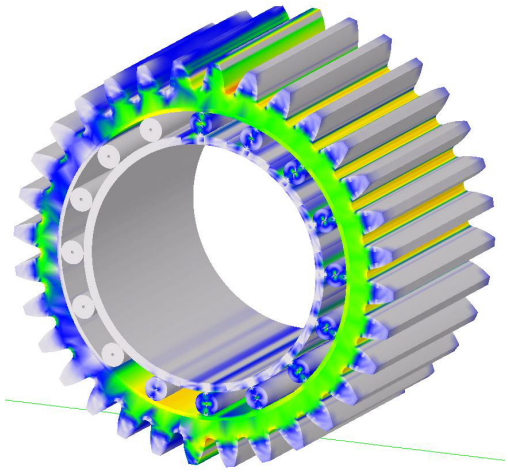

Figure 7.9: A planetary pinion with a detailed bearing model

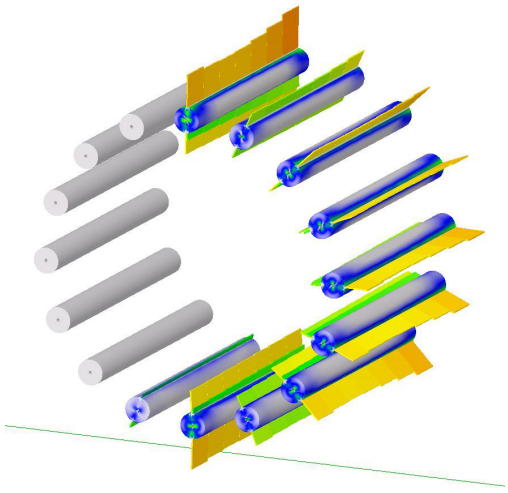

Figure 7.10: A planetary pinion with a detailed bearing model

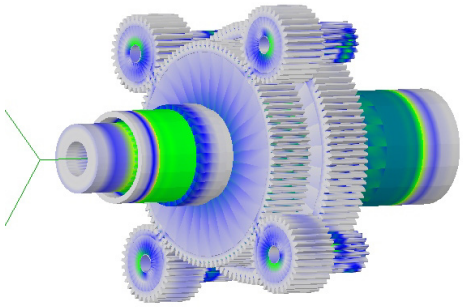

Figure 7.11: A planetary differential system model

Detailed bearing models were used between the pinion and the carrier (Figure 7.14). The deformed shape of the pinion (Figure 7.15) predicted by the model strongly depends on the fidelity of the bearing model. For thin walled pinions, the only way to correctly predict the pinion stresses is to support the pinion in this manner.

The accuracy of the deformed shape the stress predictions in the ring gear fillets also depends on the boundary conditions applied to it, as we have shown previously with the two-dimensional model. With the realistic spline supports and the realistic housing model, we obtain the ring gear shape and stresses shown in Figure 7.16.

7.6 Conclusions

Planetary3D is a very powerful package, and can be used to advantage in analyzing systems that are very difficult to model otherwise. We have already started introducing industrial users to its potential.

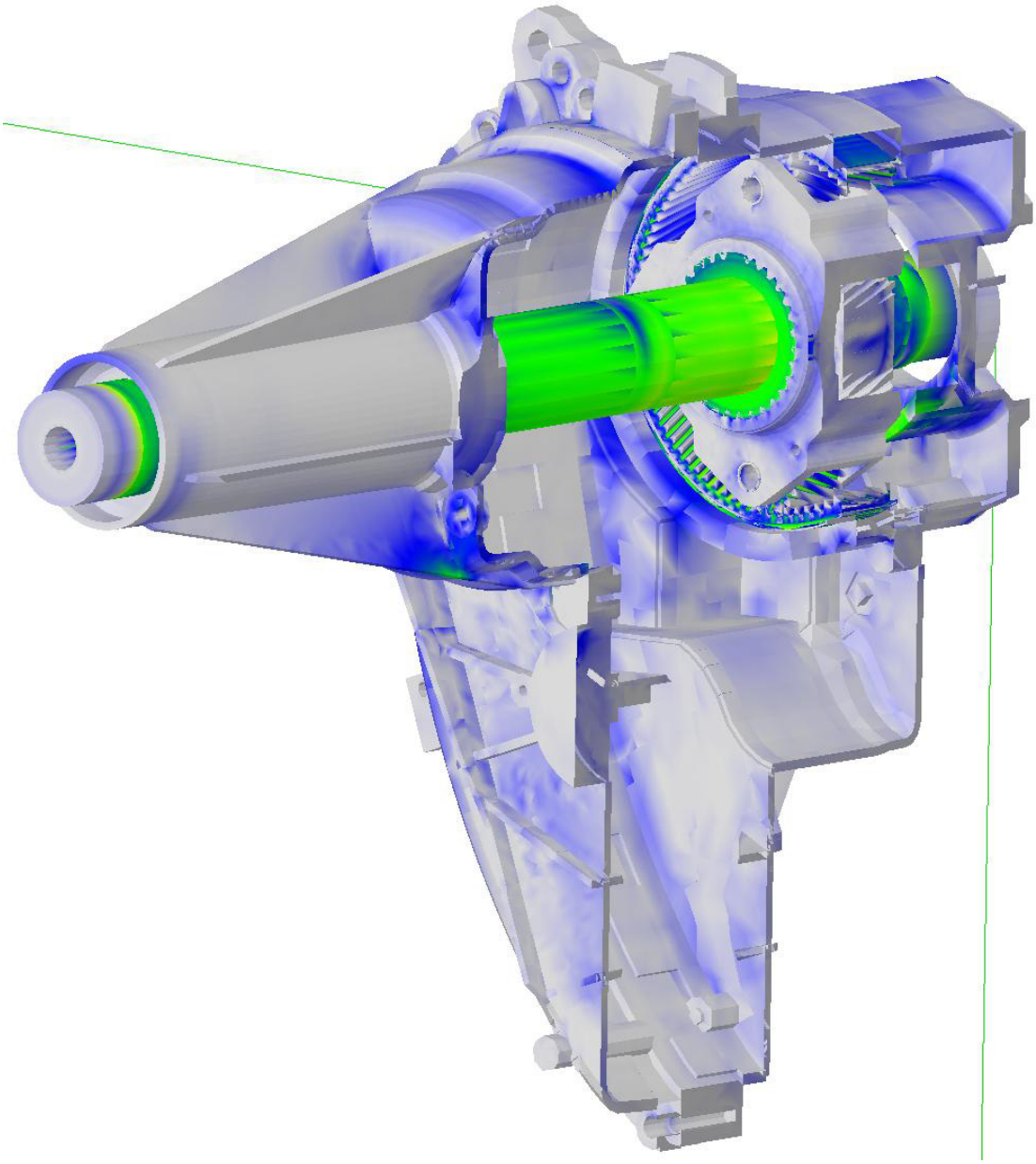

Figure 7.12: Transfer case model

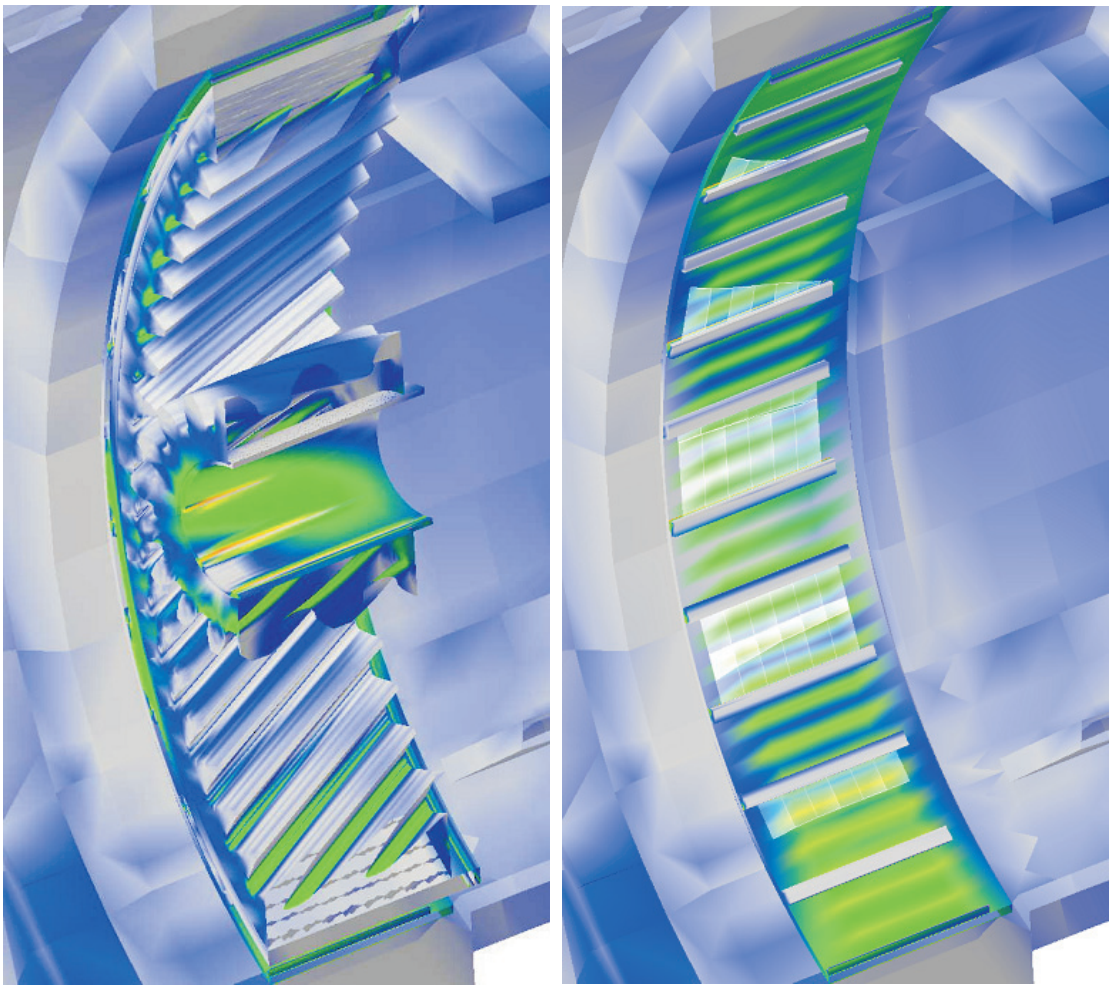

Figure 7.13: Transfer case pinion, ring and housing

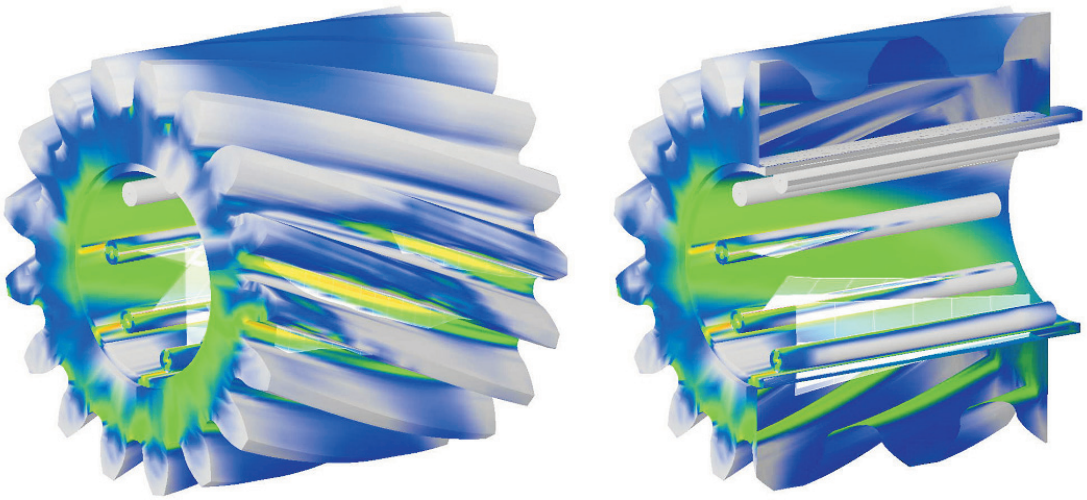

Figure 7.14: Transfer case pinion and bearing rollers

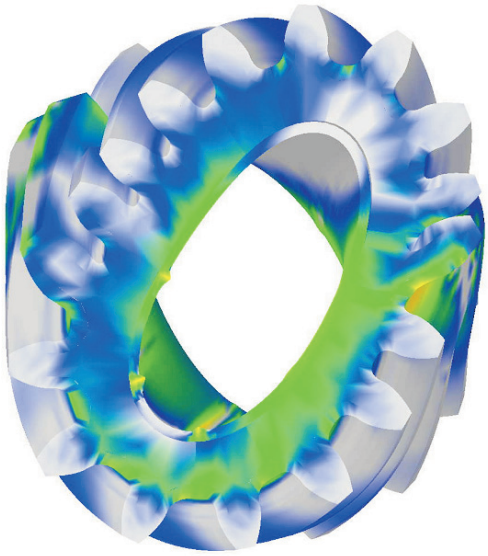

Figure 7.15: Deformed shape of the pinion in the transfer case model

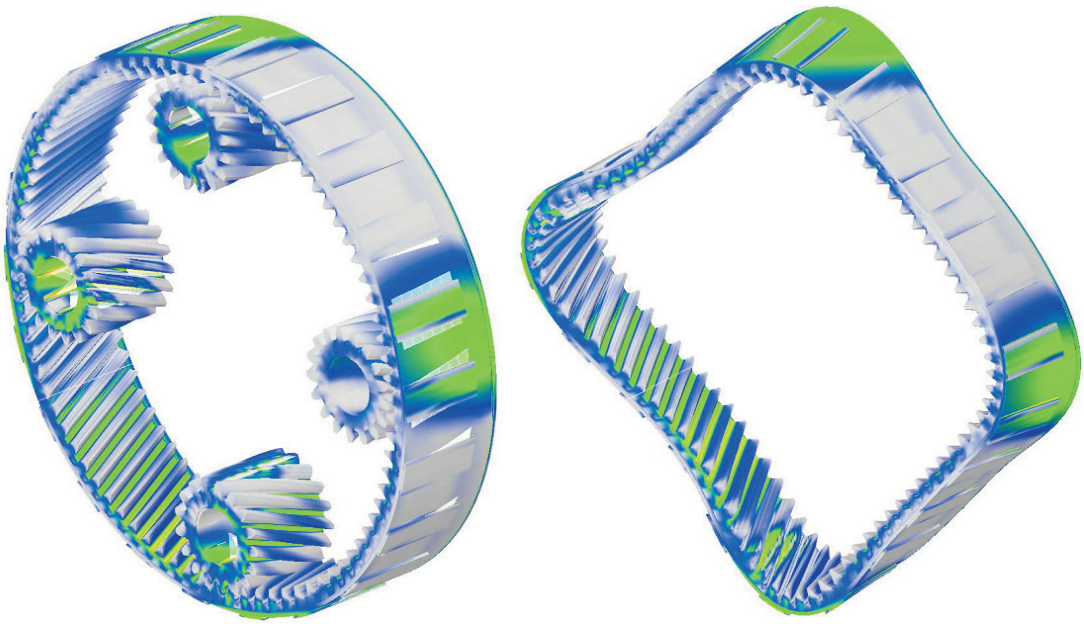

Figure 7.16: Deformed shape of the ring gear in the transfer case model

Chapter 8

Transmission3D

Our success in building and testing the *Planetary3D* package encouraged us to build an even more powerful program in *Transmission3D*.

All the features of *Planetary3D* are retained in *Transmission3D*, so we will not repeat those here. A detailed description of *Transmission3D* can be found in it user's manual [45], and in its validation manual [46].

We decided to do away with the restriction that all the gears in the system have parallel axes. We also felt that it was important to be able to include arbitrary gear models generated by other packages such as *HypoidFaceMilled*, and to allow for straight bevel differentials in the model.

The *Transmission3D* model consists of rotors, housings and connectors. The rotor axis can be arbitrary, and is set up by specifying the coordinates of one point on the axis, and a unit vector pointing along it. The rotor in turn consists of an arbitrary number of shafts, carriers, and gears. The gears can be external helical (sun) gears, internal helical (ring) gears, straight bevel gears, or arbitrary gears generated by other software packages. We refer to these arbitrary gears as 'hypoid' gears in this chapter, but in reality they can be bevels, spiral bevels, or even face gears. The rotor can have any number of carriers, and the carrier can have any number of pinions. The pinions can be straight bevel pinions, or helical pinions. Straight bevel pinions are commonly used in automotive differemtials.

The straight bevel pinions and straight bevel gears are generated within *Transmission3D* using the octoid geometry described in 2.5.

We discuss two examples which illustrate the capabilities of *Transmission3D*.

8.1 Coupled Spiral Bevel and Planetary System

A combination of a spiral bevel gear set and a planetary reduction set is shown in Figure 8.1. This combination occurs frequently in helicopters, and until now, there had been no way to study them as a coupled system. This is a hypothetical system we built as a proof of concept.

The power from the engine comes into the system through the input shaft and the spiral bevel pinion mounted on it. This input pinion mates with a spiral bevel gear, which is integral with a spur sun gear. The sun gear mates with four pinions, which in turn mate with the ring gear. The power flows out through the carrier.

For the sake of simplicity, We chose to keep the outer diameter of the ring gear rigid in this model. The carrier is also rigid, and thus needs no finite element model.

Figure 8.2 shows the stress distribution over the model. Figure 8.3 shows the stresses in the pinions, and their deformed shapes. Figure 8.4 shows the contact loads and tooth bending stress in the spiral bevel pinion on the input shaft.

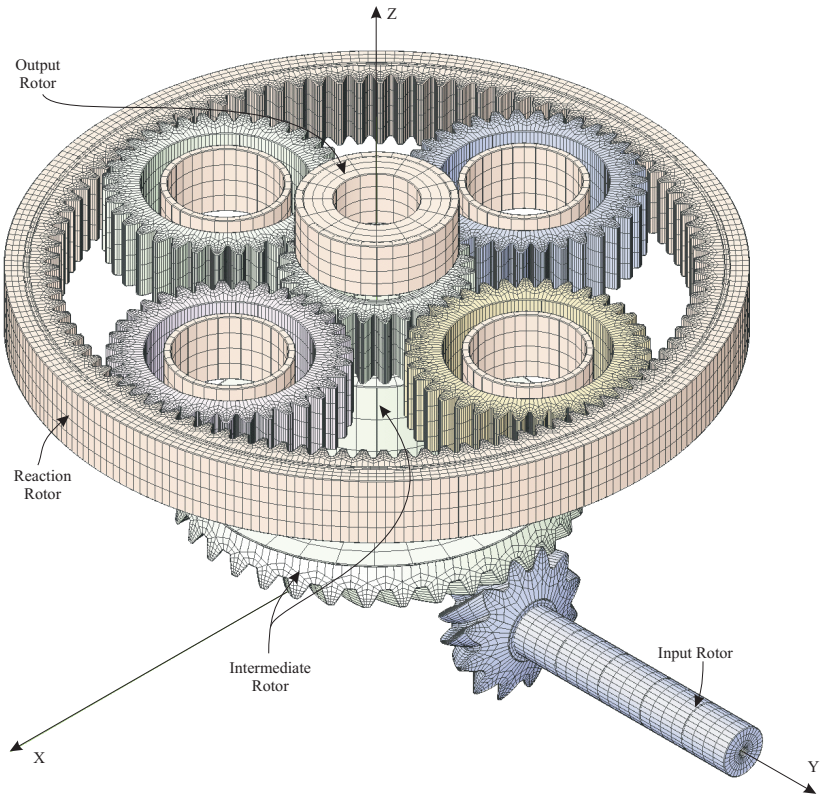

Figure 8.1: Coupled spiral bevel and planetary system.

Figure 8.2: Coupled spiral bevel and planetary system.

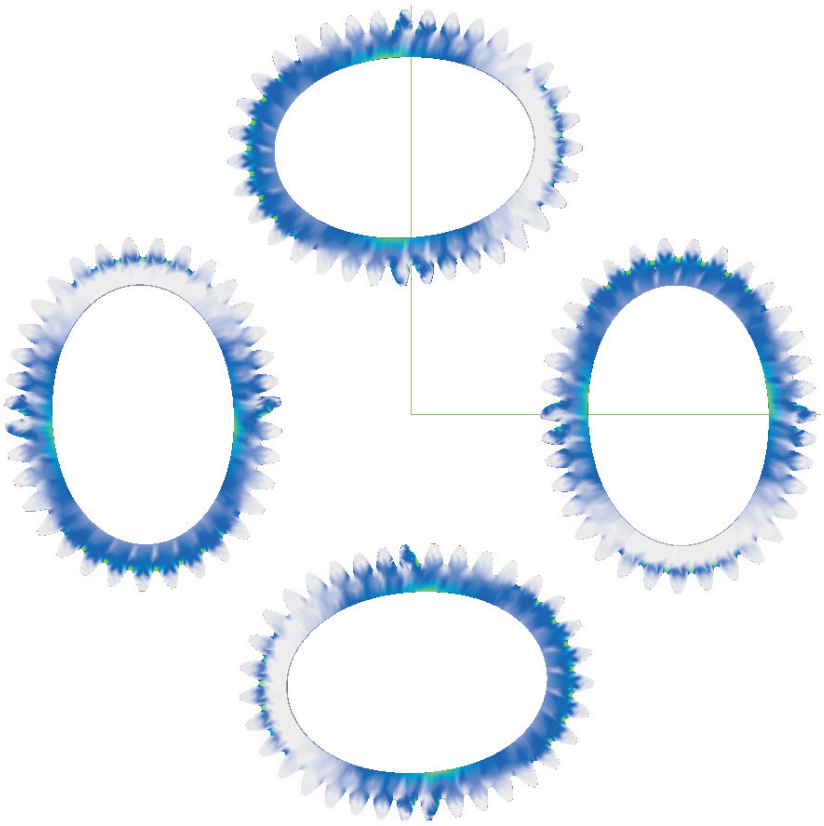

Figure 8.3: The deformed shapes of the spur pinions in the coupled spiral bevel and planetary system.

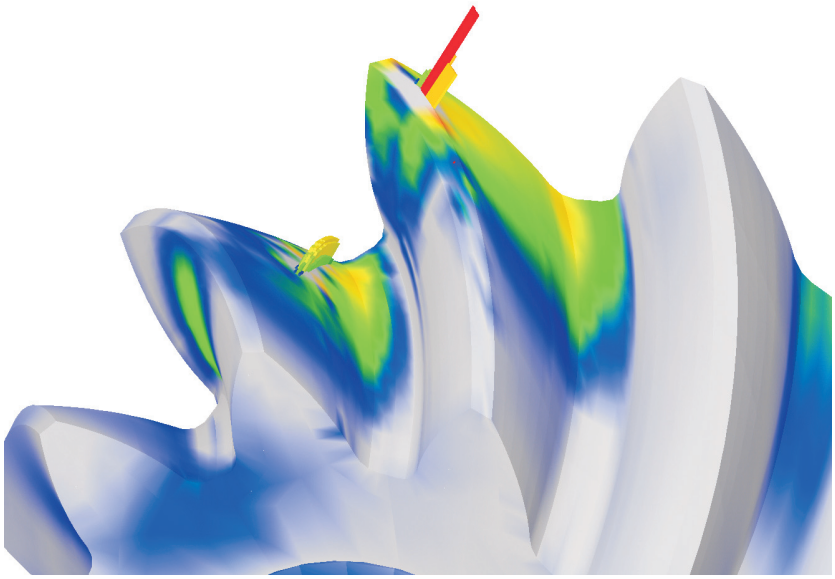

Figure 8.4: The spiral bevel pinion stresses and contact loads in the coupled spiral bevel and planetary system.

8.2 Automotive Rear Axle Gear Set

The final example we have chosen is from a heavy duty semi-tractor rear axle differential. A schematic of the rear axle system is shown in Figure 8.5. This is a complicated system with a hypoid gear set and a straight bevel differental set. The hypoid gear is integral with the differential carrier. There are four differential pinions on the carrier meshing with the straight bevel gears on two half shafts. The power flows into the system from the engine through the propellor shaft and the hypoid pinion. It flows out to the wheels through the two half shafts.

We obtained a CAD model of the housing from its manufacturer. The CAD model was in *ProE* format. We emited the surface geometry of the housing from *ProE* in *IGES* format, The surface geometry was then imported into the finite element pre-processing program *HyperMesh*. A finite element model of the housing was then created within *HyperMesh* and exported out as a *NASTRAN* bulk data file. Finally the bulk data file was converted into *Calyx* format, and then incorporated into the *Transmission3D* model. The housing model has quadrilateral and triangular shell elements, and hexahedral, pentahedral and tetrahedral solid elements. Rigid body elements were also used in the housing. The final housing finite element model in *Transmission3D* is shown in Figure 8.6.

A similar process was used to convert a CAD model of the carrier into a *Calyx* finite element model (Figure 8.7). A mesh consisting purely of tetrahedral elements was created, because of mesh generation considerations.

The hypoid gear set geometry and the straight bevel gear set geometry was not provided to us by the manufacturer, so we designed our own, possibly non-optimal, gear sets to go into this model.

Figure 8.8 shows a cutaway view of the system after analysis. Color indicate stress levels. Figures 8.9 through 8.15 show stress levels on the individual components of this system.

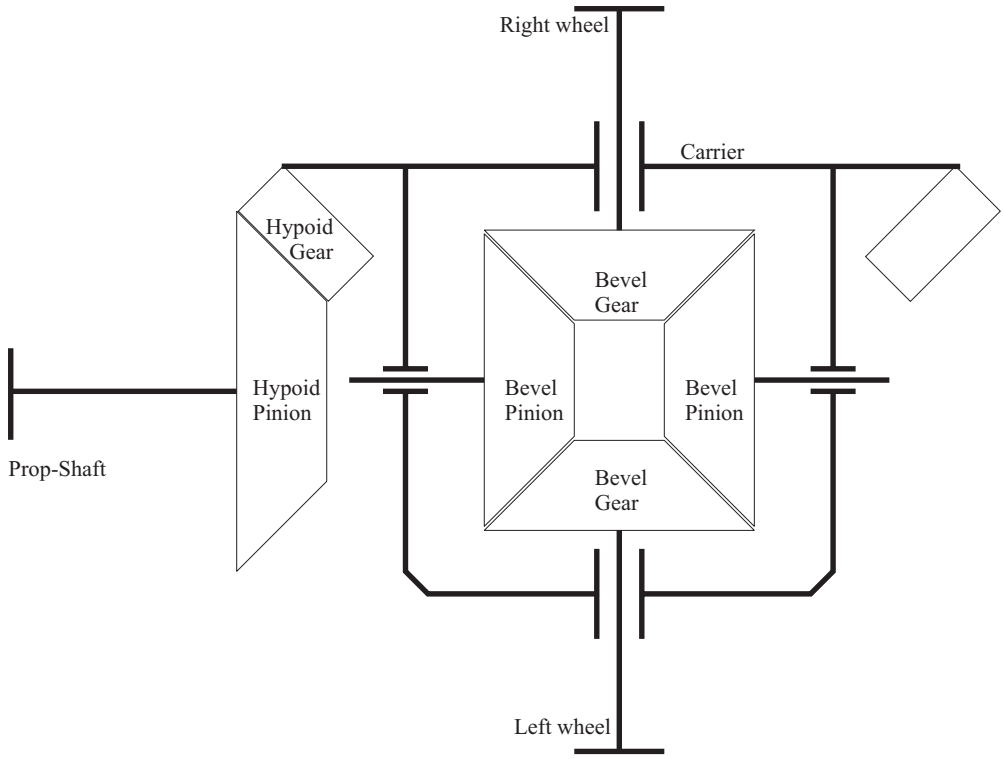

Figure 8.5: A schematic drawing of the automotive rear-axle assembly.

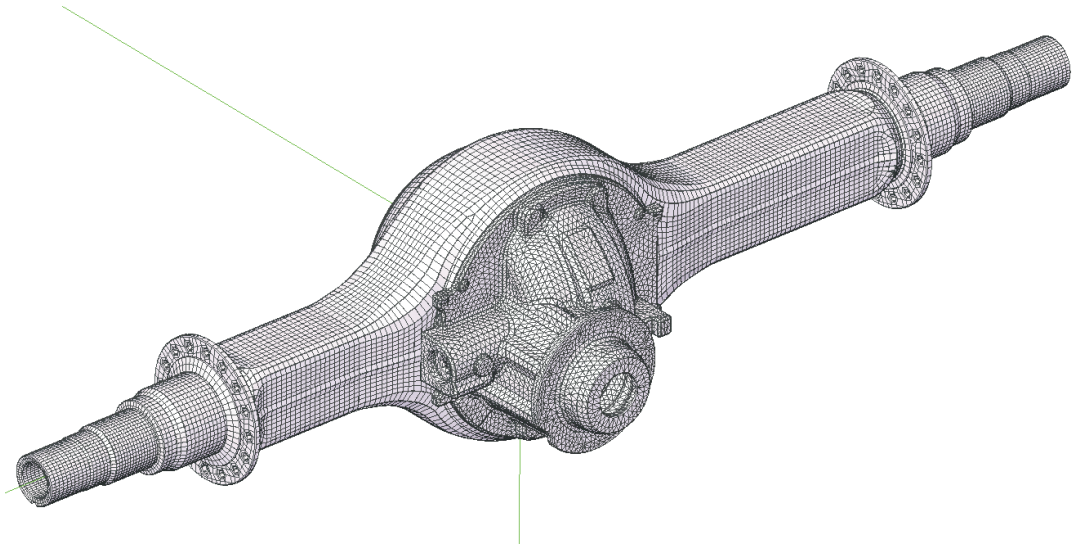

Figure 8.6: The housing finite element mesh.

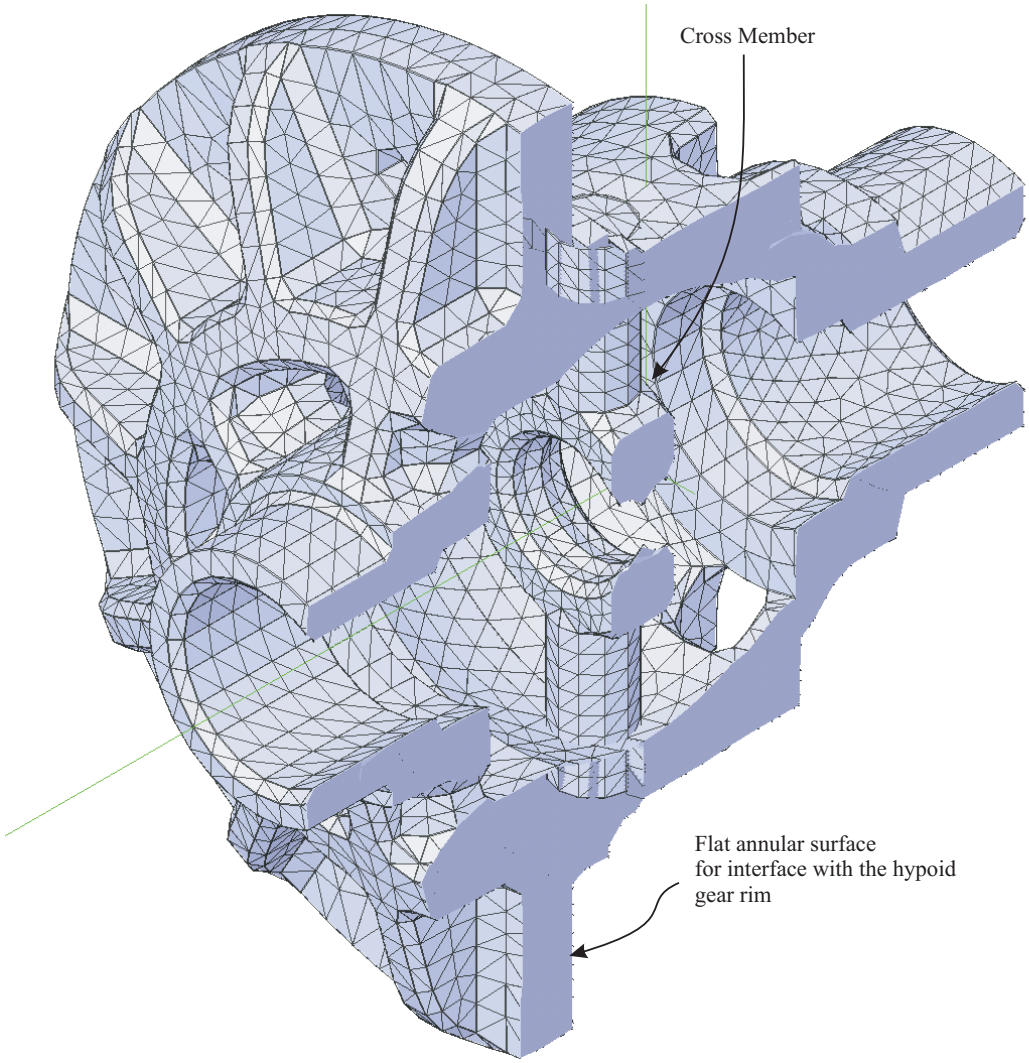

Figure 8.7: The carrier finite element mesh.

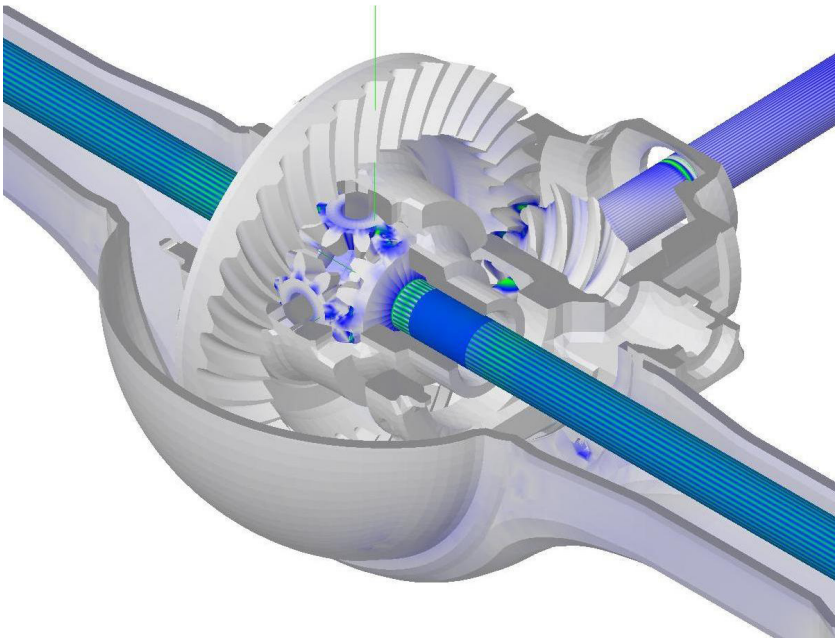

Figure 8.8: A cut-away view of an automotive rear-axle assembly.

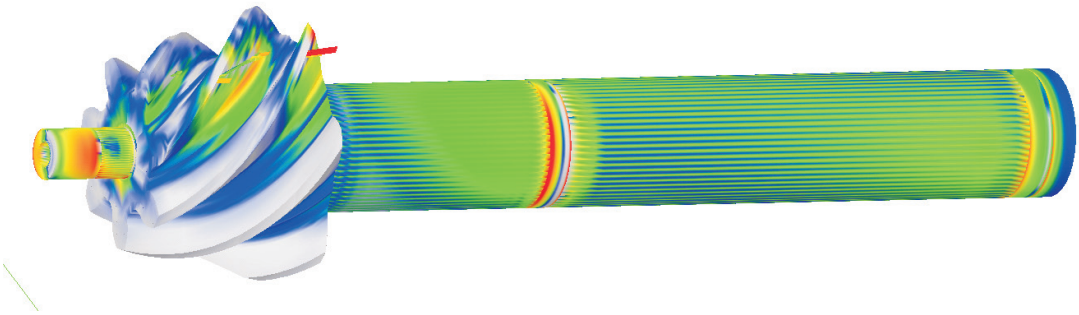

Figure 8.9: Stress contours on the hypoid pinion.

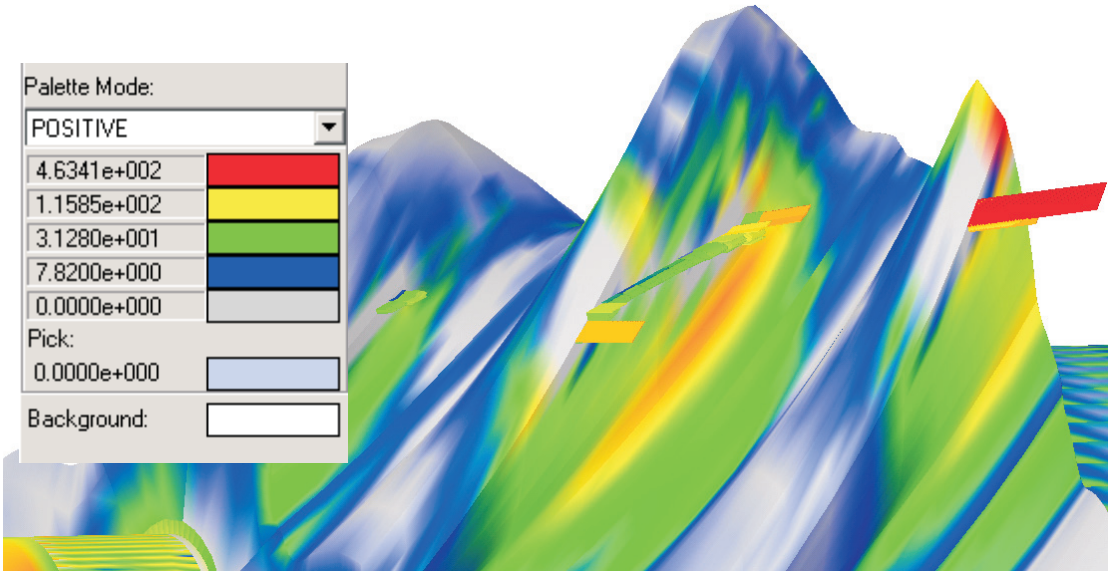

Figure 8.10: Stress contours on the hypoid pinion.

8.3 Conclusions

We have shown that *Transmission3D* can be used to analyze real world systems of considerable complexity. At the same time, since it is built on the same basic components used in other packages, and which we have extensively validated, we assured of the accuracy of its predictions.

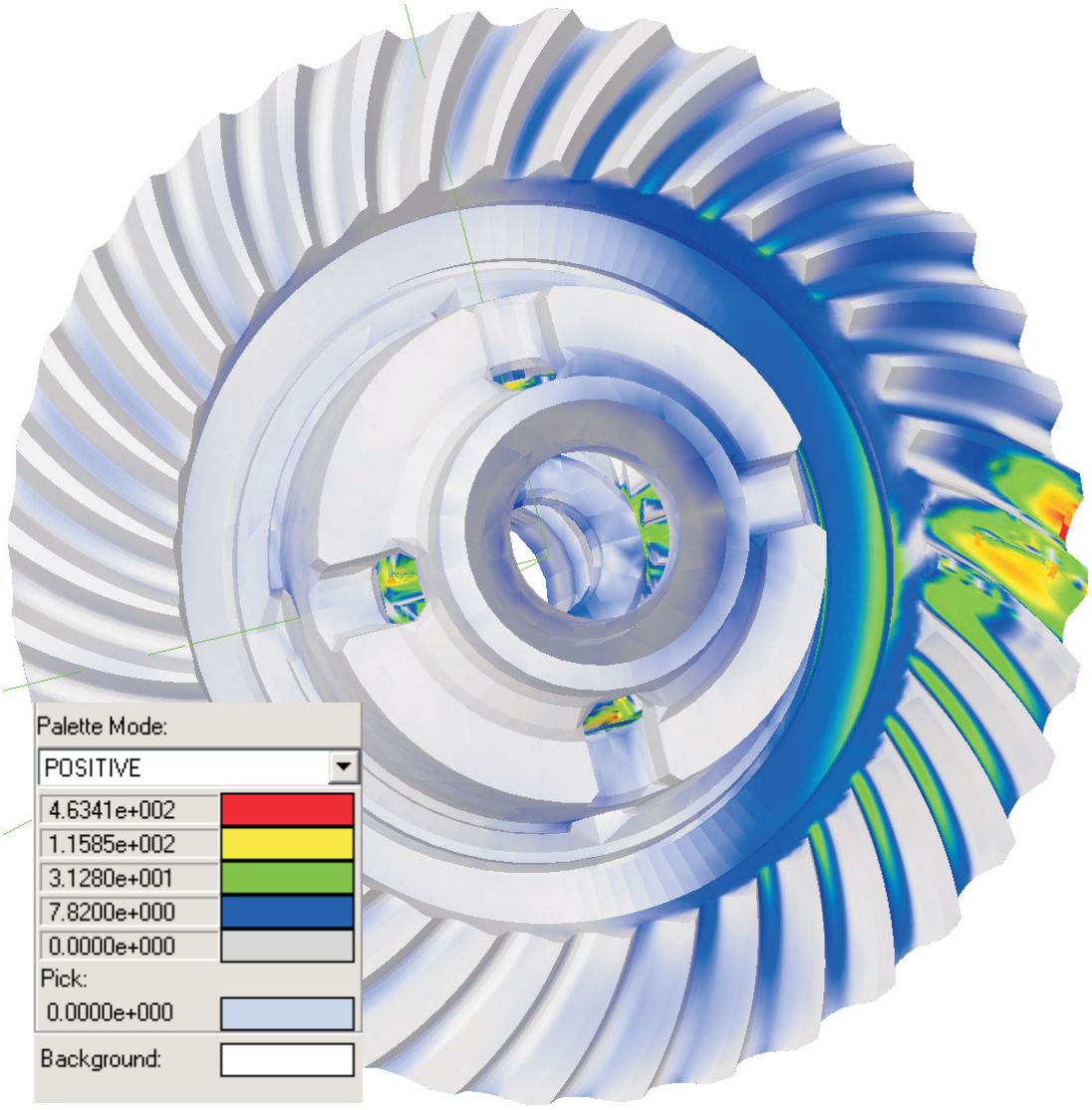

Figure 8.11: Stress contours on the hypoid gear.

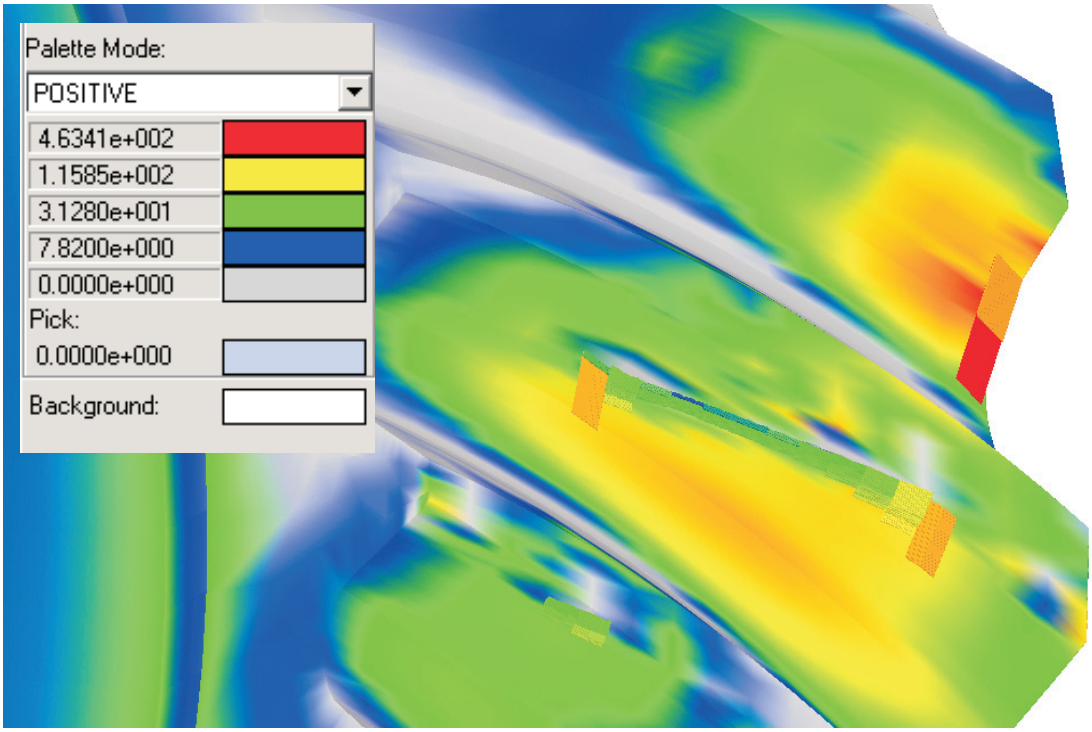

Figure 8.12: Stress contours on the hypoid gear.

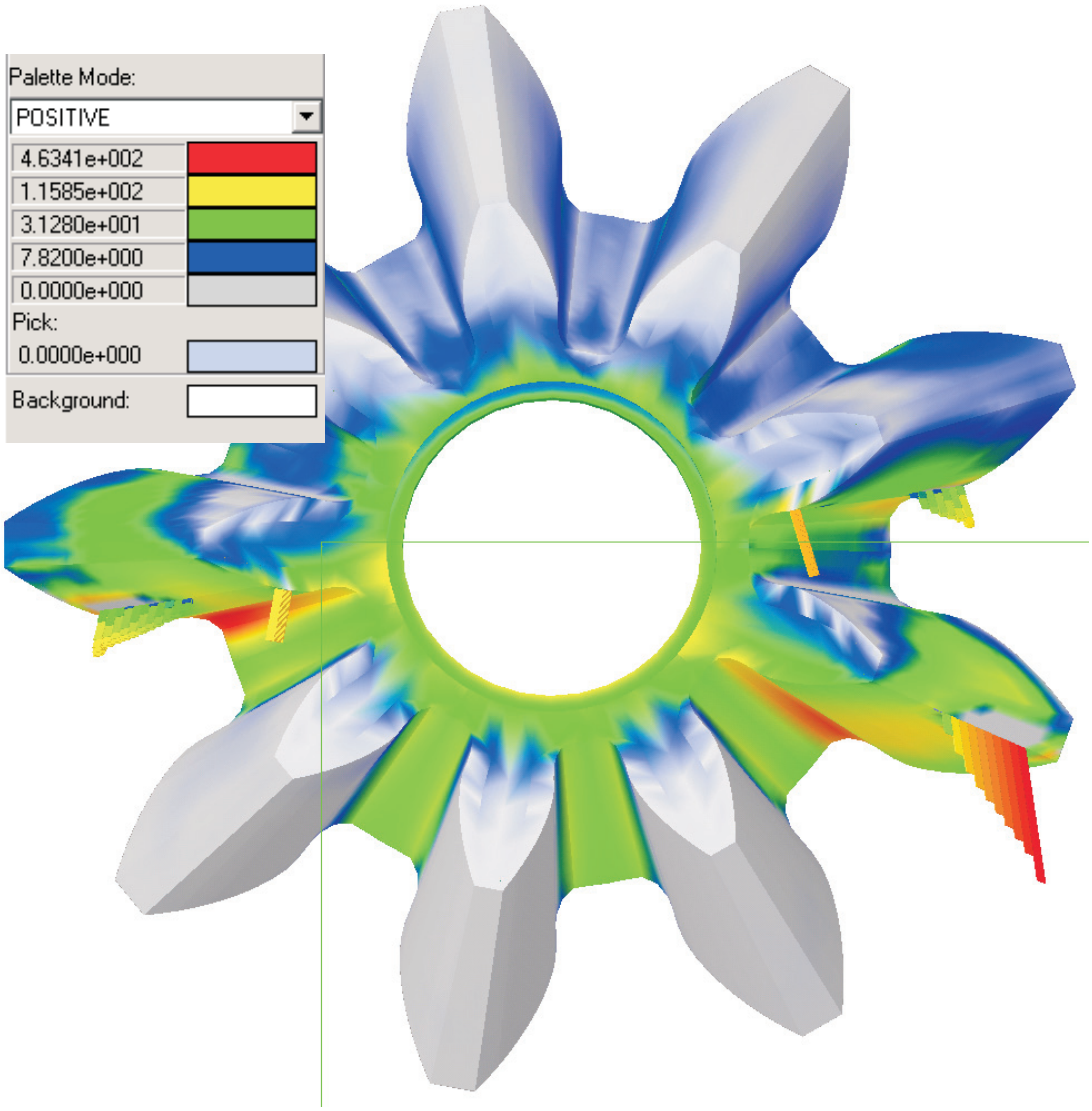

Figure 8.13: Stress contours on one of the straight bevel pinions.

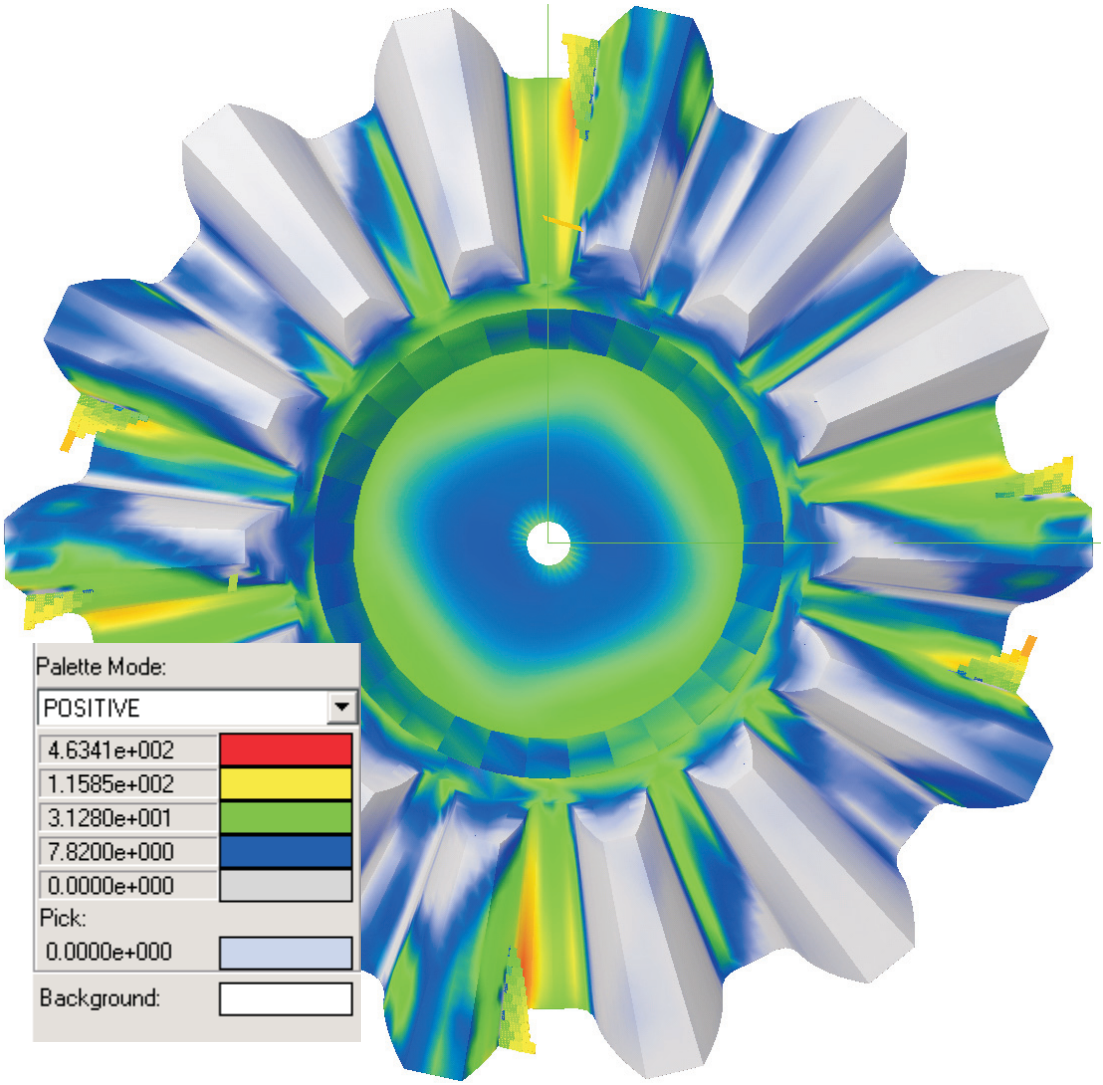

Figure 8.14: Stress contours on the right half shaft straight bevel gear.

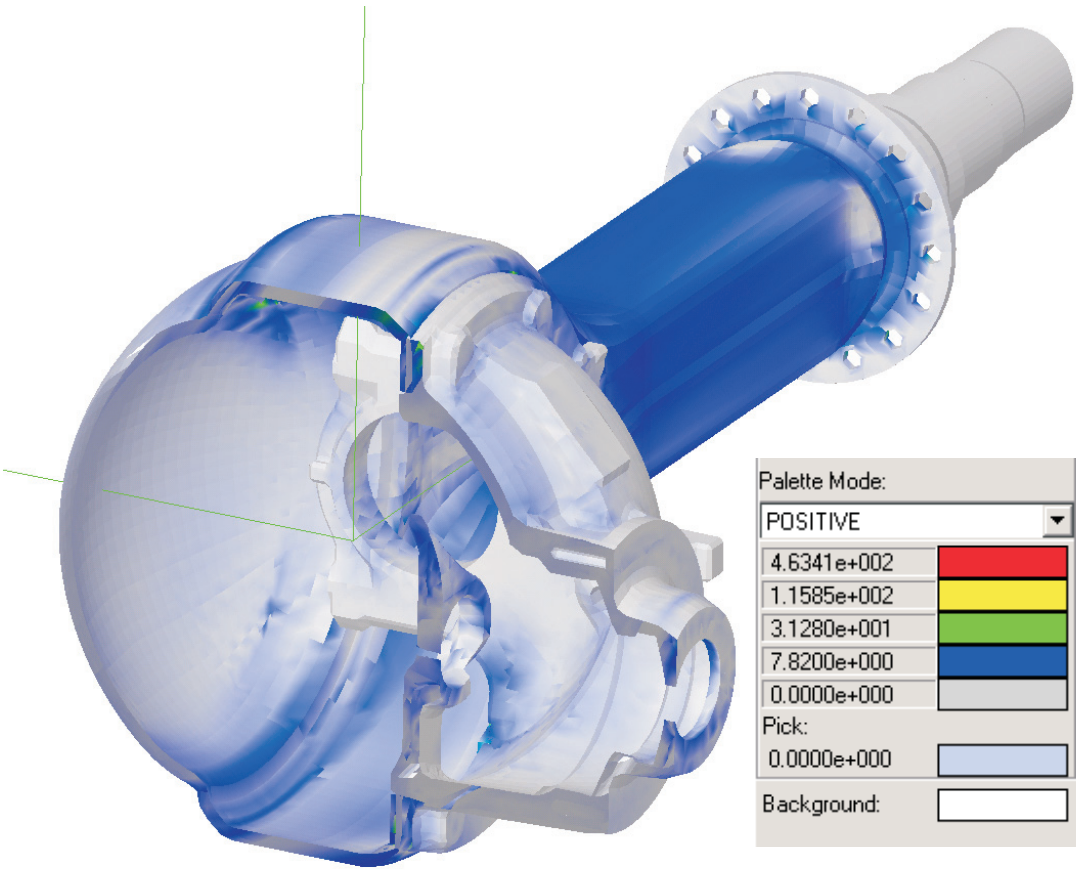

Figure 8.15: Stress contours on the housing.

Chapter 9

Conclusions and Acknowledgements

The state of the art of gear finite element analysis when we embarked on this SBIR project 36 months ago was that the best analysis tools available then could only analyze simple models with only a pair of gears and only five or six teeth on each gear. Multibody gear systems were simply out of computational reach. We had no tools available for spiral bevel and hypoid gears, other than primitive programs provided at considerable cost by the manufacturers of gear cutting machines. There was no way into include realistic bearing models and couple them with realistic gears models.

Compared to that scenario, we feel that we have significantly advanced gear analysis technology. We see promising applications in high performance and high power density aerospace and automotive applications.

We would like to acknowledge the encouragement, financial, scientific, and moral support provided by the U.S. Army Research Laboratory, and Tim Krantz in particular, without which this work would not have been possible.

Bibliography

[1] Mark Valco, Ph.D. Dissertation, *Planetary Gear Train Ring Gear and Support Structure Investigation*, Cleveland State University, 1992.

[2] Vijayakar and Houser, The use of boundary elements for the determination of the geometry factor, 1986 *AGMA Fall Technical Meeting*, Paper no. 86-FTM-10.

[3] S. Vijayakar, H. Busby and D. Houser, Linearization of multibody frictional contact problems, *Computers and Structures*, vol. 29, no. 4, pp. 569-576, 1987.

[4] S. Vijayakar, A combined surface integral and finite element solution for a three-dimensional contact problem, *International Journal for Numerical Methods in Engineering*, vol.31, pp. 525-545, 1991.

[5] S. Vijayakar, Edge effects in gear tooth contact, *ASME* 7^{th} *International Power Transmissions and Gearing Conference*, San Diego, October 1996.

[6] R. G. Parker V. Agashe and S. M. Vijayakar, Dynamic response of a planetary gear system using a finite element/contact mechanics model. *Journal of Mechanical Design*, 122:304–310, September 2000.

[7] George Hadley, *Linear programming*, Addison Wesley, 1962.

[8] Katta G. Murty, John Wiley, *Linear and Combinatorial Programming*, 1976 ISBN: 0-471-57370-1.

[9] G. Hadley, *Nonlinear and dynamic programming*, Addison Wesley Publishing company, 1964.

[10] *The Finite Element Method*, O. C. Zienkiewicz, Third Edition, McGraw-Hill Book Company, 1979.

[11] *Finite Element Procedures in Engineering Analysis*, Klaus-Jrgen Bathe, Prentice Hall, 1982.

[12] *Theory of Plates and Shells*, S. Timoshenko and S. Woinkowsky-Krieger, Second Edition, McGraw-Hill Book Company, 1959.

[13] Earle Buckingham, *Analytical Mechanics of Gears*, First Edition, McGraw Hill Book Company, New York 1949.

[14] Y. C. Tsai and P. C. Chin, Surface Geometry of Straight and Spiral Bevel Gears *Journal of Mechanisms, Transmissions, and Automation in Design, Transactions of the ASME*, 109:443–449, December 1987

[15] *Housing Noise Radiation: Theory Manual*, Advanced Numerical Solutions.

[16] *Housing Noise Radiation: Validation Manual*, Advanced Numerical Solutions.

[17] A. Seybert, T.W.Wu, and X. F.Wu. *Experimental validation of .nite element and boundary element methods for predicting structural vibration and radiated noise. Technical report*, NASA Contractor Report 4561, 1994.

[18] *Calyx User's Manual*, Advanced Numerical Solutions.

[19] S. Vijayakar, H. Busby and D. Houser, Finite element analysis of quasi-prismatic structures, *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 1461-1477, 1987.

[20] Krantz, T. L., *Gear Tooth Stress Measurements of Two Helicopter Planetary Stages*, NASA Technical Memorandum 105651, AVSCOM Technical Report 91-C-038, 1992.

[21] Blankenship and Kahraman, *Gear Dynamics Experiments, Part I: Characterization of Forced Response*, ASME 7^{th} International Power Transmissions and Gearing Conference, San Diego, October 1996.

[22] Blankenship and Kahraman, *Gear Dynamics Experiments, Part II: Effect of Involute Contact Ratio*, ASME 7^{th} International Power Transmissions and Gearing Conference, San Diego, October 1996.

[23] Blankenship and Kahraman, *Gear Dynamics Experiments, Part III: Effect of Involute Tip Relief*, ASME 7^{th} International Power Transmissions and Gearing Conference, San Diego, October 1996.

[24] Ajit Bodas and Ahmet Kahraman, Influence of carrier and gear manufacturing errors on the static planet load sharing behavior of planetary gear sets, *MPT-2001 Conference Fukuoka, Japan, September 2001*,

[25] P. Bajpai, A. Kahraman and N. E. Anderson A surface wear prediction model for parallel axis gear pairs, *to be presented at ASME/AGME 2003 International Power Transmission and Gearing Conference, Chicago, September 2003*,

[26] A. Kahraman, A. A. Kharazi and M. Umrani A deformable body dynamic analysis of planetary gears with thin rims, to appear in *Journal of Sound and Vibration*,

[27] A. Kahraman, S. Vijayakar Effect of internal gear flexibility on the Quasi-static behavior of a planetary gear set, *Journal of Mechanical Design, Transactions of the ASME*, 123:408–415, September 2001

[28] J. Lin and R. J. Parker. Analytical characterization of the unique properties of planetary gear free vibration. *Journal of Vibration and Acoustics*, 121:316–321, July 1999.

[29] R. G. Parker. A physical explanation for the effectiveness of planet phasing to suppress planetary gear vibration. *Journal of Sound and Vibration*, 236(4):561–573, 2000.

[30] M. Botman, Vibration Measurements on Planetary Gears of Aircraft Turbine Engines, *AIAA Journal*, vol. 17, no. 5, 1980.

[31] F. Cunliffe, J. D. Smith, and D.B. Welbourn, Dynamic Tooth Loads in Epicyclic Gears, *J. Eng. Ind. Trans. ASME*, May 1974.

[32] A. Kahraman, S. Vijayakar, Effect of Internal Gear Flexibility on the Quassi-Static Behavior of a Planetary Gear Set, *Transactions of the ASME*, September 2001.

[33] Jian Lin and Robert Parker, 1998 *Natural Frequency Spectra and Vibration Modes of Planetary Gears*, ASME Design Engineering Technical Conference, September 1998, Atlanta Georgia.

[34] R.G. Parker, S.M. Vijayakar and T. Imajo Non-linear Dynamic Response of a Spur Gear Pair: Modeling and Experimental Comparisons, *Journal of Sound and Vibration* (2000) 237(3):435–355

[35] J. P. Den Hartog. *Mechanical Vibrations*. Dover Publications, Inc., New York, 1984.

[36] *Helical3D User's Manual*, Advanced Numerical Solutions.

[37] *Helical3D Validation Manual*, Advanced Numerical Solutions.

[38] *HypoidFaceMilled User's Manual*, Advanced Numerical Solutions.

[39] *HypoidFaceMilled Validation Manual*, Advanced Numerical Solutions.

[40] Robert F. Handschuh, Comparison of Experimental and Analytical Tooth Bending Stress of Aerospace Spiral Bevel Gears, *4th World Congress on Gearing and Power Transmission*, Paris France March 1999

[41] *Planetary2D User's Manual*, Advanced Numerical Solutions.

[42] *Planetary2D Validation Manual*, Advanced Numerical Solutions.

[43] *Planetary3D User's Manual*, Advanced Numerical Solutions.

[44] *Planetary3D Validation Manual*, Advanced Numerical Solutions.

[45] *Transmission3D User's Manual*, Advanced Numerical Solutions.

[46] *Transmission3D Validation Manual*, Advanced Numerical Solutions.