LUND UNIVERSITY | LUND INSTITUTE OF TECHNOLOGY
Division of Structural Mechanics | Sweden 1998 | Report TVSM-3035

METHODS FOR AIRCRAFT NOISE AND VIBRATION ANALYSIS

MATS GUSTAVSSON

LUND UNIVERSITY | LUND INSTITUTE OF TECHNOLOGY
Division of Structural Mechanics | Sweden 1998 | Report TVSM-3035
CODEN: LUTVDG / (TVSM-3035) / 1-81 / (1998) | ISSN 0281-6679

METHODS FOR AIRCRAFT NOISE AND VIBRATION ANALYSIS

MATS GUSTAVSSON

Preface

This licentiate thesis is a result of a research project within the national aeronautics research programme (in swedish: Nationella Flygtekniska Forskningsprogrammet, NFFP). The financing and the acceptance for changing the original time schedule to allow a fruitful interaction between this project and the BRITE/EURAM project ASANCA II is gratefully acknowledged. The experimental data used for this thesis was acquired within the ASANCA II project and made available by Saab AB. The permission from Saab AB and the other ASANCA II partners to use these experimental results is gratefully acknowledged.

The major objective of this work was to study practical methods for analysis of low-frequency noise and vibration problems, in particular for turbo-prop aircraft applications.

The inspiration for this work comes, to a significant part, from Urban Emborg, Saab AB, and William Halvorsen, Halvorsen Associates. They introduced me to the field of aircraft noise and vibration characterization and reduction, and have been of great support during the course of work leading to this thesis. In addition Mr. Halvorsen converted my version of English into proper American English (except for Paper A; thanks to Henry Rice for improving this part), which I highly appreciate.

For the practical arrangement of my post-graduate studies and for the help with computer resources I thank my supervising professor Göran Sandberg, division of Structural Mechanics, Lund University.

Finally I would like to thank my colleagues and friends at Saab AB and the Division of Structural Mechanics for their inspiration and help during this work.

Background

Aircraft noise control engineering is a challenge both for experimentally based methods and for numerical analysis methods. Usually noise control installations are designed based on engineering judgement in combination with results from in-flight testing, rather than with nummerical methods, due to the absence of reliable and accurate models.

For turbo-prop aircraft, such as the Saab 340 and Saab 2000, the low-frequency tonal noise generated by the propellers is a major reason for passenger discomfort. Two effective methods of reducing this propeller noise are tuned vibration absorbers/dampers and active noise control. The efficiency of both of these two methods is, to a great extent, determined by the placement of the damper/absorber or active actuator (structural exciter or loudspeaker). Finding appropriate locations for such installations from experiments requires extensive testing, both in-flight and on ground. If alternative design methods requiring less, or ideally no, testing could be utilized this would allow significant cost reduction and the possibility of concurrent design for the noise control installations.

That is the motivation for this study on methods for aircraft acoustic and structural vibration analysis.

Objective

The objectives of the work presented in this thesis were to:

- *evaluate the U-vector Expansion Method (UEM) using experimental data for a fully furnished aircraft section.*
- *use experimental results to "tune" numerical models for acoustic and vibration analysis*
- *investigate the possibility of using alternatives to the modal sub-structuring technique for efficiently solving large acoustic and structural dynamics numerical models.*

Each task is addressed in a separate paper. A common ground for all three papers is the ground vibration and acoustic tests performed as a part of the BRITE/EURAM project ASANCA II using a Saab 340 acoustic test section.

The emphasis has been on practical methods for analysis of aircraft acoustics and vibration analysis. For more detailed theoretical discussions the reader is referred to the references given in the end of each paper.

Results summary

In the first paper, “Dynamic modeling of a turbo-prop aircraft using the U-vector Expansion methods”, it is shown that the U-vector expansion method can be used to form acoustic and structural dynamics analysis models for the Saab 340 based on measured frequency response functions at the propeller blade passage frequency.

The second paper, “Inverse modeling of the dynamic properties of a turbo-prop aircraft”, describes the use of inverse modeling for low-frequency structural vibrations. The geometrically very simple finite element model of the fuselage structure is capable of giving properties reasonably similar to the experimentally derived dynamic properties.

Finally, the third paper, “Evaluation of Solution Methods for Finite Element Analysis of Vibrations and Acoustics”, points out the possibility of using iterative methods when solving large equation systems resulting from finite element formulation of acoustic and structural dynamic systems. In particular, the iterative conjugate gradient method is shown to be more efficient than direct solvers for large acoustic problems. For the structural dynamics examples the iterative solver converges only when using pre-conditioning. The present implementation of the pre-conditioned conjugate gradient solver cannot compete with the highly optimized direct solvers for structural dynamics models.

Dynamic modeling of a turbo-prop aircraft using the U-vector Expansion Method

Mats O. Gustavsson
Division of Structural Mechanics, Lund University, Sweden

ABSTRACT: Dynamic modeling of a furnished turbo-prop aircraft is performed by the use of a recently introduced technique called the U-vector Expansion Method (UEM). Transfer functions derived from test data are used as input for the modeling and the resulting model consists of a similar, but significantly larger, set of transfer functions to be used for design of noise reducing measures.

1. Introduction

Noise and vibration problems in turbo-prop aircraft are often characterized by the high contribution of tonal components generated by the rotating propeller blades. Typically the noise level in the passenger cabin of a turbo-prop aircraft, with no special noise reducing measures, is in the range 80-90 dB(A), totally dominated by the propeller Blade Passage Frequency (BPF) component. Reducing the tonal noise at the BPF normally results in a reduction of the total noise level by 5-10 dB(A), bringing the noise level down to a more acceptable level.

The noise control methods used for this type of low frequency noise are typically tuned vibration absorbers or/and active noise control. Both of these two methods require models with a large number of possible 'source positions' i.e the mounting points for the tuned vibration absorber units, or loudspeaker positions for the case of using active noise control. It is also desirable to be able to predict the response at a large number of positions, including the passenger ear positions. To get the dynamic properties of the fuselage structure-cabin cavity system for this number of driving points and response locations requires extensive testing, with a high cost associated.

The U-vector expansion method may be an attractive way to reduce the amount of testing to be performed, or to extend the test data base to get a model with more 'driving points' than actually used in the test.

2. The U-vector expansion method

The U-vector expansion method (UEM) [1,2,3] is based on measured transfer function data and is used to model dynamic systems. The method was developed as a result of poor performance of Experimental Modal Analysis (EMA) for heavily damped systems in frequency ranges with fairly high modal density, typical properties for vibro-acoustic problems in turbo-prop aircraft.

UEM is essentially based on the concept of singular value decomposition of a transfer function matrix, and may be seen as a natural step after the introduction of principal component analysis and the complex modal indicator function. Both principal component analysis and the complex modal indicator function are, as well as UEM, based on the concept of singular value decomposition of a matrix [4].

From a mathematical point of view UEM is nothing but a method to re-construct a symmetric, rank deficient matrix, from a set of vectors spanning the total vector space of the matrix. The rank deficiency makes it possible to use a sub-set of a matrix to find a set of space vectors spanning the space of the complete matrix. This is the central part of the UEM. The symmetry property of a matrix means that the same space vectors can be used to span both the row space of the matrix and the column space of the matrix. This is the second key characteristic on which the UEM is founded.

By using the concept of singular value decomposition a symmetric matrix, $\boldsymbol{A}$, may be decomposed in

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{s}\boldsymbol{U}^{\boldsymbol{T}} \qquad \textbf{[eq. 1]}$$

$\boldsymbol{U}$: is a matrix with orthogonal column vectors

$\boldsymbol{s}$: is a diagonal matrix

$[*]^T$: denotes the 'ordinary' transpose of a matrix (simply interchanging rows and columns).

The column vectors in ***U*** are called the *U*-vectors, and the diagonal elements in **s** are the singular values of ***A***.

The number of non-zero singular values in **s** is equivalent to the rank of ***A*** and consequently if ***A*** is rank deficient only some of the *U*-vectors are required to generate the complete matrix ***A***.

Consider if only some of the columns in the matrix ***A*** are available. This part of ***A*** is denoted $\overline{\boldsymbol{A}}$. If the number of columns in $\overline{\boldsymbol{A}}$ is equal to, or higher, than the rank of the complete matrix ***A*** it possible that the columns of $\overline{\boldsymbol{A}}$ span the column space of the complete matrix ***A***. The more columns in $\overline{\boldsymbol{A}}$, compared to the rank of ***A***, the more likely this situation becomes.

If the *U*-vectors of $\overline{\boldsymbol{A}}$ span the same column space as the *U*-vectors of the complete matrix ***A***, it is possible to re-construct the complete matrix ***A*** using the U-vectors of $\overline{\boldsymbol{A}}$. A similar relation as [Eq. 1] can be formed, but since now we do not necessarily have the same *U*-vectors as would be found from the complete matrix ***A***, a general matrix Σ, has to be used in place of the diagonal matrix **s**.

$$\boldsymbol{A} = \boldsymbol{U}(\overline{\boldsymbol{A}})\Sigma\boldsymbol{U}(\overline{\boldsymbol{A}})^{T} \qquad \textbf{[eq. 2]}$$

The matrix Σ can be determined by using $\overline{\boldsymbol{A}}$,

$$\overline{\boldsymbol{A}} = \boldsymbol{U}(\overline{\boldsymbol{A}})\Sigma\overline{\boldsymbol{U}}(\overline{\boldsymbol{A}})^{T} \qquad \textbf{[eq. 3]}$$

with $\overline{\boldsymbol{U}}(\overline{\boldsymbol{A}})$ being a sub matrix of $\boldsymbol{U}(\overline{\boldsymbol{A}})$ with the entities to get the columns of $\overline{\boldsymbol{A}}$. The matrix Σ is typically non-diagonal, but symmetric to ensure the symmetry of ***A***.

Relating the mathematical description above to the case of modeling dynamic systems, the matrix ***A*** could be a transfer function matrix ***H(f)*** at a particular frequency ***f***. The entities in the transfer function matrix $\boldsymbol{H}_{ij}(\boldsymbol{f})$ gives the response at location ***i*** for a unit input at ***j***. By using complex notation both the amplitude and phase of the relation between input and response (output) is given by $\boldsymbol{H}_{ij}(\boldsymbol{f})$.

The symmetry requirement for the complete matrix ***H(f)*** is equivalent to assuming reciprocity for the dynamic system. This means that interchanging driving point and response point would give the same relation between excitation and response - i.e the transfer functions $\boldsymbol{H}_{ij}(\boldsymbol{f})$ and $\boldsymbol{H}_{ji}(\boldsymbol{f})$ are identical.

The assumption of a rank-deficient matrix is, for the case of transfer function matrices, related to the number of modes active at the frequency of interest. For the dynamic systems considered here the modal density is high and, consequently, several significant 'principal response components' (U-vectors) must be used to get accurate models. This implies that several driving points have to be used in the test.

3. Test Description

The test case used to evaluate the performance of UEM is to model the dynamic behavior of a turbo-prop aircraft. Transfer function measurements on a 2.5 m long Acoustic Test Section of the Saab 340 are used as the input data for the UEM.

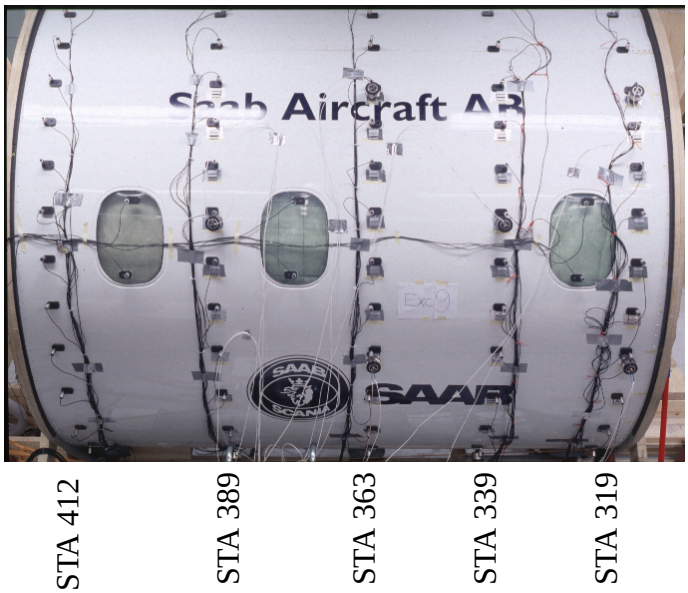


Figure 1: The Saab 340 Acoustic Test Section.

In the test 140 accelerometers and 105 microphones were used to register the acceleration response of the structure (Figure 1,2) and the acoustic response in the cabin cavity (Figure 3). The locations are identified by the cross-section name ('STA' in figure 1) and the cross-sectional position (figure 2,3).

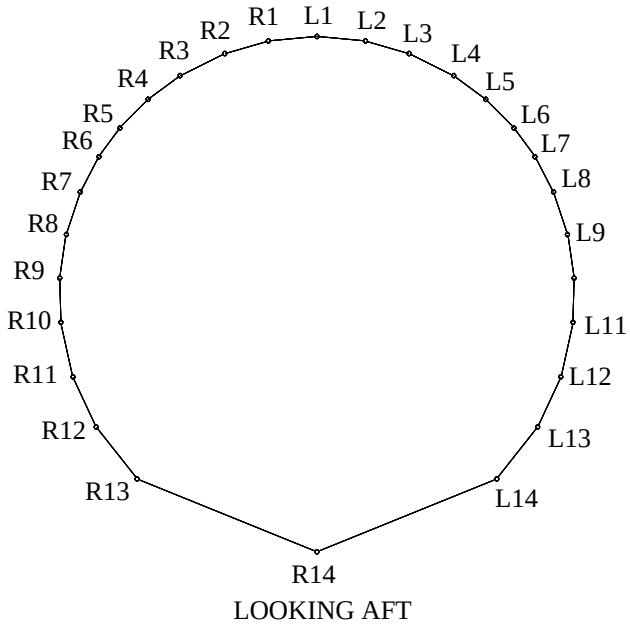


Figure 2: Positions used for vibration measurements and structural exciters.

What makes this test quite unique is the high number of excitation positions used: 108 structural excitation locations and 65 acoustic excitation positions.

For the structural excitation electro-dynamic inertia shakers were used, and for the acoustic excitation small loudspeakers were used.

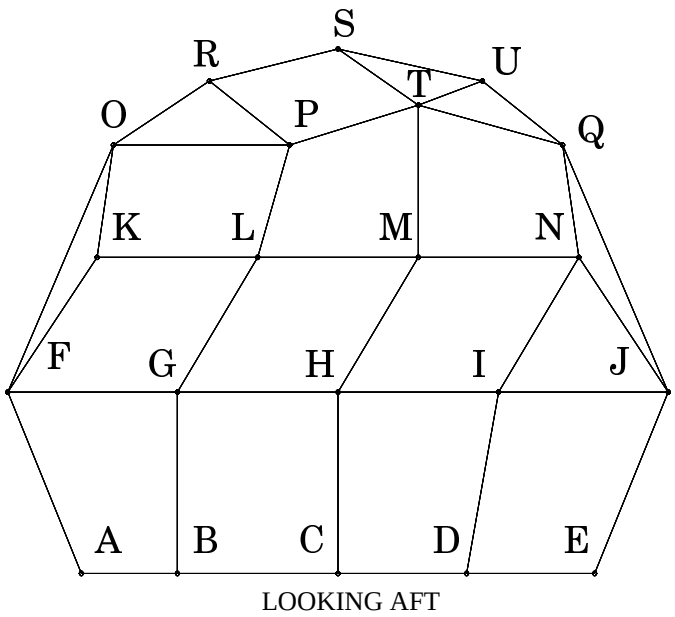


Figure 3: Positions used for acoustic pressure measurements and loudspeakers.

4. Transfer function data

In order to get reliable models with the U-Vector Expansion Method it is essential to remember the assumption of a:

- Rank deficient

and

- Symmetric

transfer function matrix.

Working with test data, and quantities derived from test data, one has to accept some deviations from the two assumptions above in the original data. It is, however essential to force the transfer function matrix to be symmetric before applying UEM.

Forcing symmetry is equivalent to eliminating factors that cause non-reciprocal data. In the current data set the response measurements were taken not exactly at the point of the excitation but at a distance of approximately 5cm from the excitation point for the structural locations and at about 10 cm from the 'source centre' for the loudspeaker excitation. This could not be avoided due to installation restrictions (i.e the size of the exciter/loudspeaker and the actuators). It is believed that this misalignment is more serious in some positions than others, in particular for the structure, explaining the difference in reciprocity characteristics.
Another reason for the in some cases poor reciprocity is the fact that the response level differs significantly for the two types of excitation (structural vs. acoustical). For the structural excitation the structural response was about 0.05 m/s^2 and the acoustic response was about 20 mPa. For the case of acoustic excitation the response levels were about 0.02 m/s^2 and 500 mPa respectively. With this quite large difference in response levels it is likely that non-linearity effects are influencing the results. In particular the attachment of the interior trim panel is a source of non-linear behavior.

To estimate the validity of the reciprocity assumption a spatial correlation coefficient, ***sac***, for two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is used.

$$sac = \frac{uv^H}{\|u\|\|v\|} \qquad \textbf{[eq. 4]}$$

Table 1 give the average correlation for structural/ structural acoustic/acoustic and structure/acoustic (or visa versa) degrees of freedom (dof's) at the frequency line of most interest (82 Hz).

Table 1: Reciprocity correlation

Driving point/ Response point	*sac*
Structure/Structure	0.87
Acoustic/Acoustic	0.90
Structure/Acoustic (and vice versa)	0.64

The propeller Blade Passage Frequency of the Saab 340 is 82 Hz. That is the reason why the work presented in this paper is focused on 82 Hz.

In principle UEM can be applied to combined vibration/acoustic data, but with the quite low correlation for the structural/acoustic reciprocity found in the current data set, separate models will be derived for the structure and the acoustics respectively.

5. Test data Preparation

A first step, before the UEM is applied, is to 'prepare' the transfer function matrix to obey the symmetry requirement. Several methods may be used to force symmetry. Probably the simplest method is to use the average of $\boldsymbol{H_{ij}(f)}$ and $\boldsymbol{H_{ji}(f)}$. In matrix notation

$$H(f)_{sym} = (H(f) + H(f)^T)/2 \qquad \textbf{[eq. 5]}$$

However a method based on response rank reduction might be a better choice. To fulfill the assumption of a rank deficient and symmetric transfer function matrix we may use some of the U-vectors of $\mathbf{[H(f)\, H^T(f)]}$ and let

$$H(f)_{sym} = U\bar{s}V_{avg} \qquad \textbf{[eq. 6]}$$

U: is the U-matrix of [H H^T]

$\bar{s}$: is a diagonal matrix with a selected set of the singular values of [H H^T]

V: is the average of the part of *V* related to the H and H^T respectively

Leuridan et al. [5] describe a similar method to force reciprocity of transfer functions, although their method uses data for a frequency band rather than spatial data.

The rank deficiency requirement does not have to be handled separately, but will be a natural part of the UEM.

6. Transfer function matrix expansion

With as many as 108 structural and 65 acoustical driving points available in the test data set the performance of the UEM can be evaluated starting with different sub-sets of the complete test data. In this case the transfer functions derived with a model, based on typical test data, can be compared with the actual (measured) transfer functions.

Data for 30 frequency lines is available but, again, the major interest is to derive models valid at the fundamental Blade Passage Frequency (BPF) of the Saab 340, which is 82 Hz.

7. Results

Starting with a transfer function matrix containing all structural responses and all but one of the driving points (i.e using data for 107 of the 108 driving points) the 'missing' transfer function is derived by UEM. Comparing this result to the actual (measured) transfer function give an estimate of how well unmeasured transfer functions can be modeled with the existing data. Figure 4, 5 show two comparisons between measured and UEM-modeled transfer functions at 82 Hz using 25% of the U-vectors for the UEM.

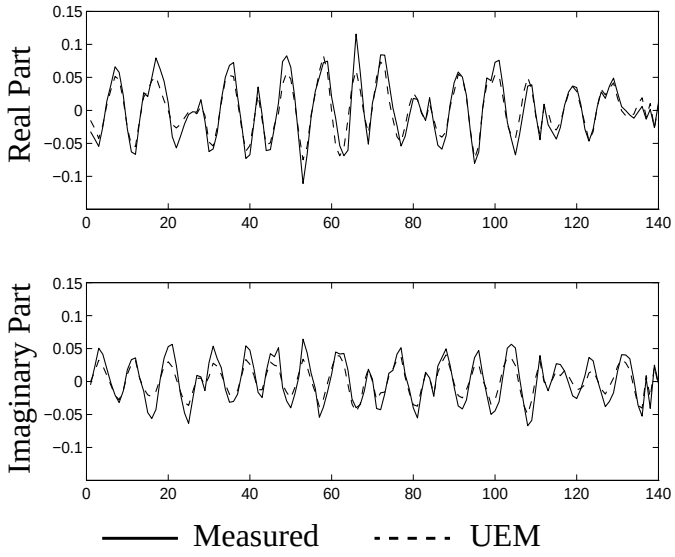


Figure 4: Structural transfer function response. Excitation at STA 363 pos. R10.

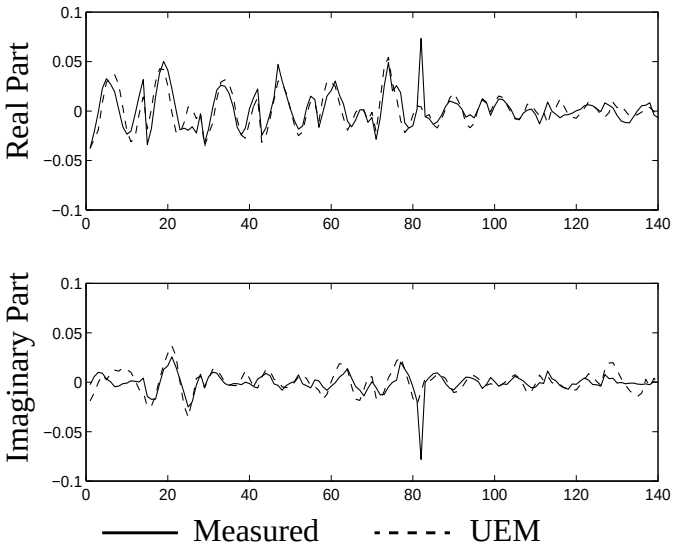


Figure 5: Structural transfer function response. Excitation at STA 339 pos. R12.

Figure 6,7 show a 'snap-shot' of the measured and UEM derived structural response respectively.

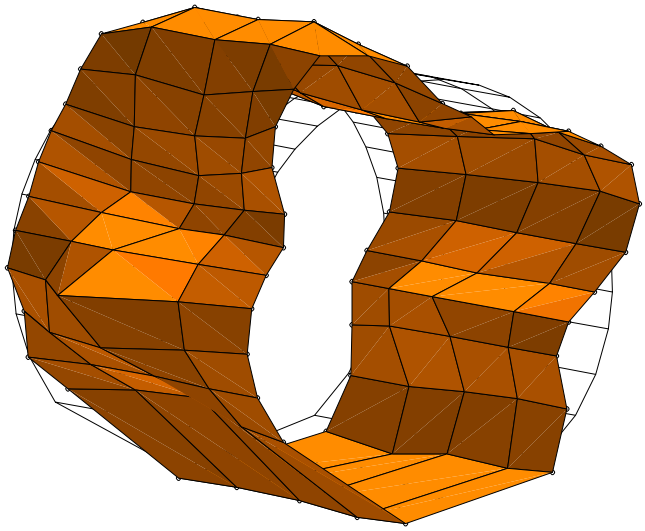


Figure 6: Measured transfer function response. Excitation at STA 363 pos. R10.

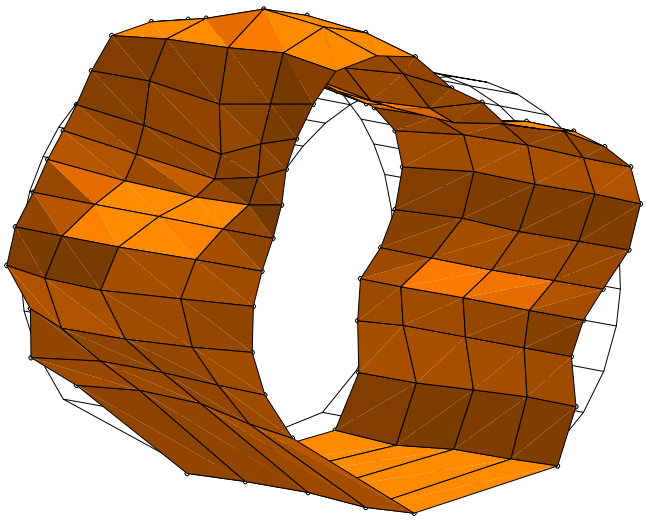


Figure 7: UEM transfer function response. Excitation at STA 363 pos. R10.

The result in Figure 4-7 is for the case of using 26 of the 107 available U-vectors (25%) for the UEM, which is equivalent to assuming a transfer function matrix with rank 26. The average of the correlation coefficient, defined by [eq. 4], for the measured and modeled transfer function is 0.82.

The effect of using different portions of the U-vectors is given by table 2.

Table 2: Correlation for the structural model (107 of 108 driving points used).

U-vectors	10%	25%	50%	100%
Correlation	0.77	0.82	0.78	0.69

For this particular set of data it appears to be optimal to use about 25% of the 107 U-vectors.

For the acoustic data similar calculations were made. In this case there are data for 65 driving points (loudspeaker locations), and using the same portion of the U-vectors (25% 50% and 100%) is equivalent to using 16, 32 and 64 U-vectors. The results in figure 8,9 are for the case of using 25% of the U-vectors.

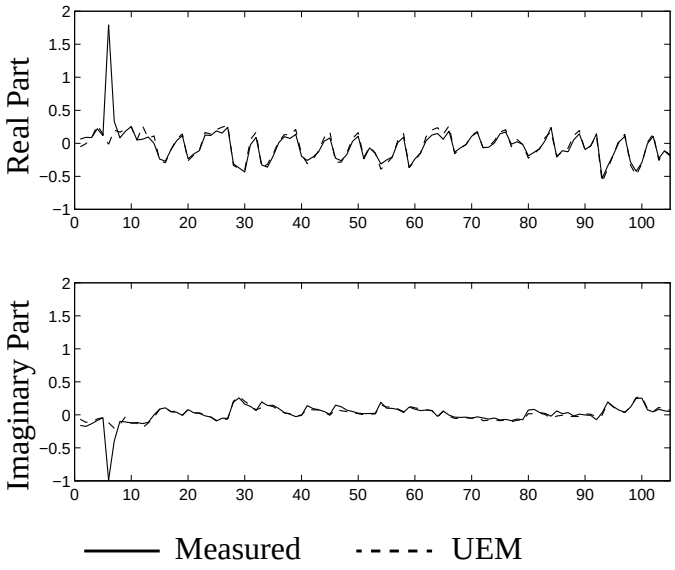


Figure 8: Acoustic transfer function response. Excitation at STA 319 pos. A.

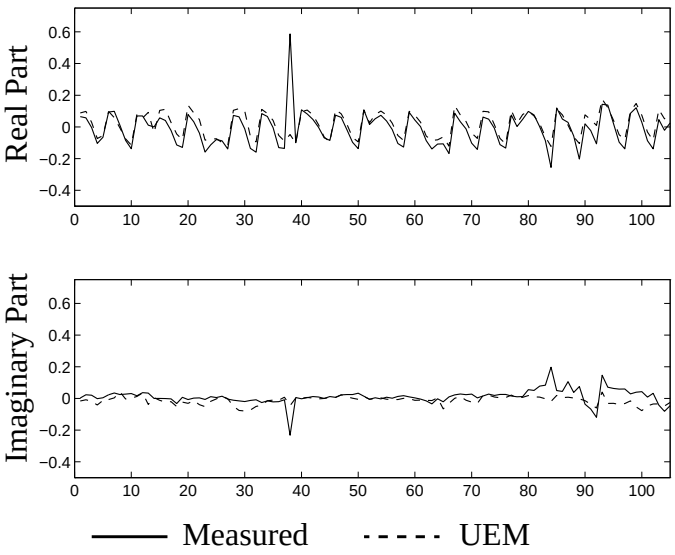


Figure 9: Acoustic transfer function response. Excitation at STA 389 pos. E.

From table 3 one gets the impression that the results for the acoustic data are not as good as for the structural data, but taking a closer look at Figure 8,9 this is found to be a consequence of the expected poor performance of the UEM for the driving point response.

Table 3: Correlation for the acoustic model (64 of 65 driving point used).

U-vectors	10%	25%	50%	100%
Correlation	0.57	0.58	0.58	0.30

This is typical for expansion methods based on principal response vectors (UEM), or modal vectors (EMA), if the driving point response is high compared to the response at other points in the system. Excluding the driving point response, the result is completely different (Table 4).

Table 4: Correlation for the acoustic model excluding the driving point.

U-vectors	10%	25%	50%	100%
Correlation	0.80	0.79	0.84	0.42

In figure 10, 11 show the measured and modeled transfer function response respectively for acoustic excitation at STA 319 pos. A.

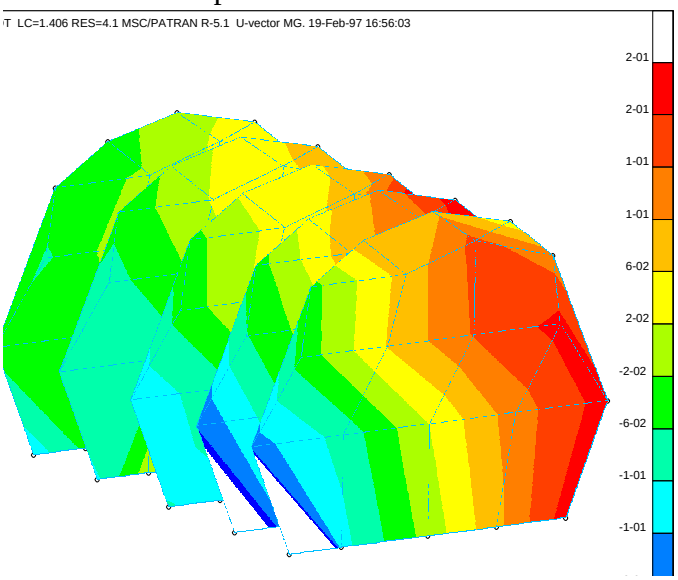


Figure 10: Measured transfer function response. Excitation at STA 319 pos. A.

The correlation between the measured and modeled response is quite good, although the measured response close to the driving point differs.

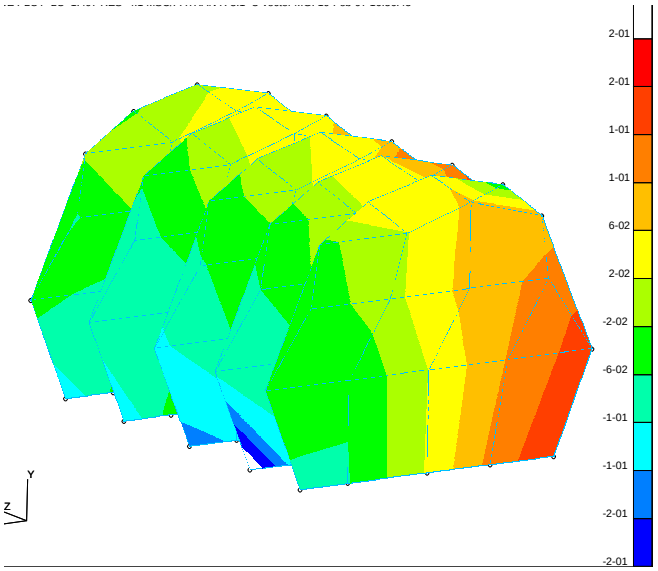


Figure 11: UEM transfer function response. Excitation at STA 319 pos. A.

The results above are for the case of taking a very large part of the data and using it to derive a model. For the structural excitation case 108 of the 140 columns of the transfer function matrix were available, and in the acoustic data 65 of the 105 columns were present. A more realistic situation would be to use data for significantly fewer driving points; however the number of driving points has to be large enough to allow for finding a set of U-vectors spanning the space of the complete transfer function matrix to be derived by UEM.

A 50% reduction of the number of driving points would be a significant reduction in test time, both for the data acquisition and for the installation time for structural exciters/loudspeakers. To simulate this situation only 50% of the transfer function data available was used for the UEM. Again we can compare the UEM results with the 'true' transfer functions as they were determined by the test.

For the structural data almost the same results as when using all but one transfer function were obtained (Table 5.).

Table 5: Correlation of the structural model using 50% of the transfer functions.

U-vectors	10%	25%	50%	100%
Correlation	0.71	0.80	0.77	0.76

Not only the average correlation but also the individual responses appear to by only slightly influenced by the quite substantial reduction in input for the UEM (figure 12,13).

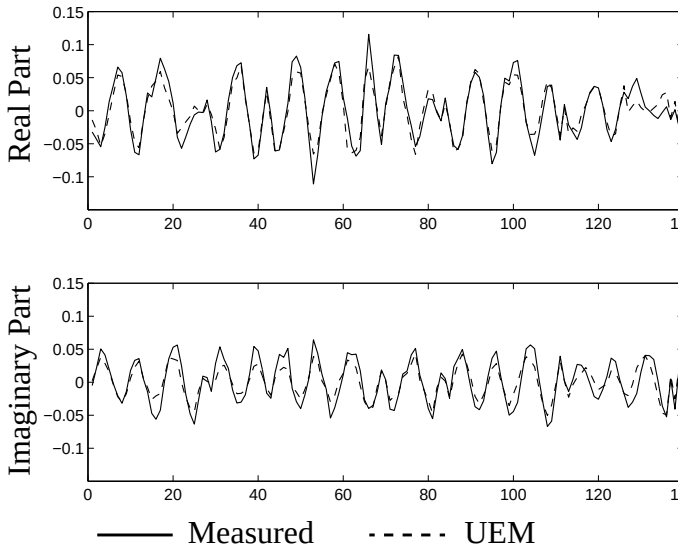


Figure 12: Structural transfer function response. Excitation at STA 363 pos. R10.

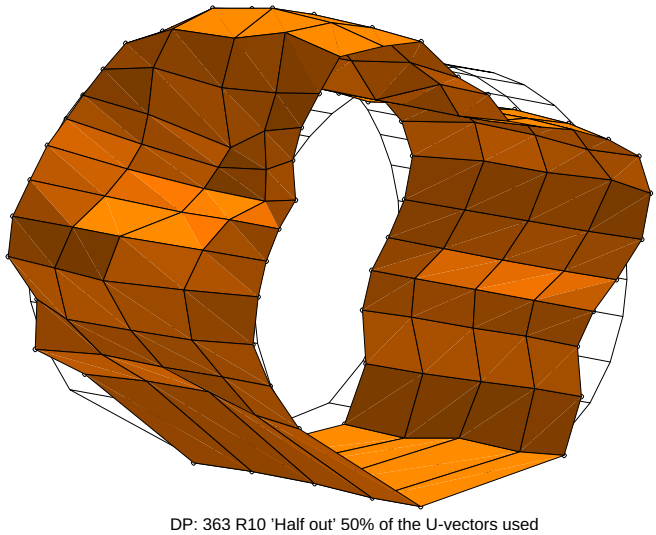


Figure 13: UEM transfer function response. Excitation at STA 363 pos. R10.

Also for the acoustic data no significant performance degradation of the UEM is found when using only half the existing transfer functions for the modeling (Table 6, Figure 14,15).

Table 6: Correlation for the acoustic model excluding driving point response and using 50% of the transfer functions.

U-vectors	10%	25%	50%	100%
Correlation	0.74	0.80	0.79	0.61

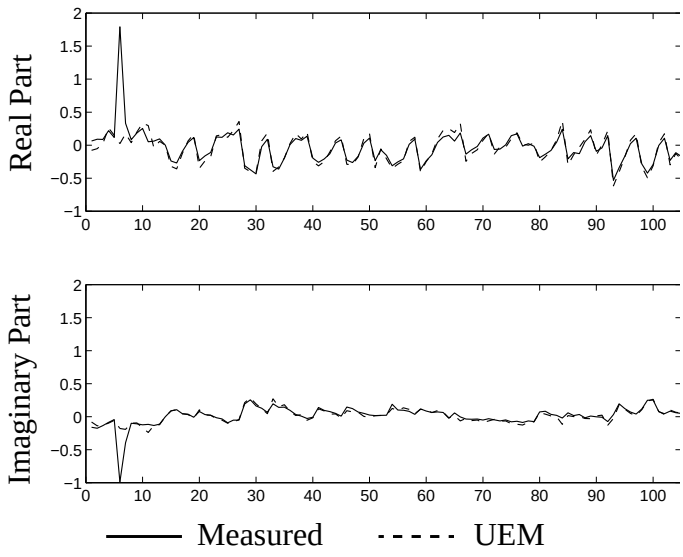


Figure 14: Acoustic transfer function response. Excitation at STA 319 pos. A.

The good result for the UEM using 50% of the transfer function data encourages one to try using fewer of the available transfer functions. The results will, however, now become very sensitive to which transfer functions (driving points) are used to model the response of a certain driving point.

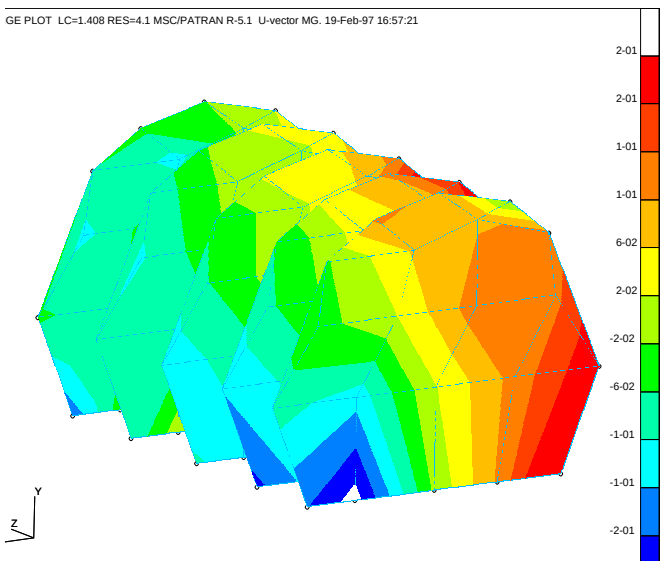


Figure 15: UEM transfer function response. Excitation at STA 319 pos. A.

8. Conclusions

The U-vector Expansion Method (UEM) has been used to model the structural vibrations and the acoustic response of a turbo-prop aircraft. The results show that quite accurate models of the structural vibration and the acoustic response can be achieved by UEM. Almost no degradation in model performance was found using 50% of the available test data compared to the case of using practically all the test data available for the modeling.

9. Acknowledgments

The work presented in this paper is financed by the National Swedish Aeronautics Research Program (NFFP). All test data used for the analysis were made available by Saab AB and are a result of the BRITE/ EURAM project ASANCA II.
The financial support by NFFP, and the permission for using test data from the ASANCA II project are gratefully acknowledged.

10. REFERENCES

[1] Gustavsson M.
A method to generate dynamic models and its application to aircraft noise.
SVIB Conference Riksgränsen, Sweden 1991.

[2] Halvorsen W, Barney P, Brown D.
The U-vector expansion method for modelling structural/acoustic systems.
Journal of Sound and Vibration, July 1991.

[3] Otte D.
Development and evaluation of singular value analysis methodologies for studying multivariante noise and vibration problems
Doctoral dissertation,
Katholieke Universiteit Leuven, Belgium

[4] Golub G.H, Van Loan C. F.
Matrix Computations, Second Edition (1989).
The John Hopkins University Press.

[5] Leuridan J. et al.
Coupling of Structures Using Measured FRF's: Some Improved Techniques.
Proc. 13th Int. Seminar on Modal Analysis,
Leuven Belgium.

Inverse Modeling of the dynamic properties of a turbo-prop Aircraft

Mats Gustavsson

Div. of Structural Mechanics, LTH, Lund University, Lund, Sweden

ABSTRACT

Inverse modeling has been used to define dynamic models of the Saab 340 acoustic test section. Ground vibration measurements performed in the BRITE/EURAM project ASANCA II constitute the dynamic properties to be reproduced by the models. The finite element technique was used with a general objective to define accurate, but computationally efficient, models rather than geometrically precise models.

In general the derived models represent the dynamic properties accurately in the frequency region of the Saab 340 blade passage frequency (82-85 Hz). This was the major purpose of the inverse modeling and the achieved results could be used for designing active or passive vibration control installations.

The deviations between the measurements and the finite element model responses are moderately small. To some extent the quality of the measurement data prohibits perfect correlation. However the simplified mass distribution used for the model is probably the major reason for the deviations.

Additionally it was found that the derived models not only capture the dynamic response at the blade passage frequency, but also to some extent the variations with frequency.

1 Introduction

Dynamic analyses of aircraft structures play a central role in cabin noise and vibration control. Numerical models based on the Finite Element Method (FEM) are attractive in the sense that they can be used at early stages of a product design process, when no measurements can be performed. Later, when test data becomes available, the models can be updated to better match the measurement results and subsequently used for evaluation of design modifications and design optimization.

It is well known that the dynamic properties of an aircraft are significantly influenced by the installation of interior components. Especially the damping characteristics are drastically changed. Consequently the numerical models have to be representative of a furnished aircraft.

Defining finite element models for dynamic analyses of aircraft fuselage structures is, however, far from trivial. This is indeed the case when furnishing components and the thermal insulation material are to be included. In the present paper an inverse modeling approach is used to derive an finite element model of the Saab 340 turbo-prop aircraft.

2 *Inverse modeling*

If the dynamic properties of a large structure are to be determined it may be both unnecessary and practically impossible to include all details in the construction. There may also be uncertainties on a local level, when it must be defined if components are in contact or not; if a connection is to be regarded as infinitely stiff or as linear or non-linear translation and rotation springs, etc. Furthermore, in some cases also the constitutive models for the materials may be difficult to determine. This certainly is the case for the thermal insulation materials, but even materials that are quite simple to model in a quasi-static sense may require a more sophisticated material description if a dynamic model is incorporated. Most important in dynamic analysis is to have a proper representation of the damping characteristics (i.e the energy losses), which normally is not a problem for quasi-static analysis. A structure may also behave quite differently at 'vibration amplitudes' compared to the behavior at larger deformations. A typical example of this are the differences found for a riveted structure compared to a welded structure with respect to dynamic properties, while the two structures may behave identically for a static load. A riveted structure normally shows significantly higher damping as relative motion between the rivet and the structural members occurs.

Naturally, it is not feasible to model each rivet of an aircraft structure; thus it makes sense to try to include the effects of the rivets, as a modified material property. Test results are in that case required to determine the new "material model", including not only the material itself, but also accounting for the effect of the rivets. Raising this approach to a higher level, the complete aircraft cabin wall may be represented with a single material model. All properties of the cabin wall can, of course, not be represented by this simplified model, but this may not be required for the application.

Assume that some global properties of a structure are available from measurements, e.g. the global response for various dynamic excitations. This can be used to determine material properties for a simplified model of the structure. Finding these material properties is a typical inverse modeling problem.

2.1 Gradient based parameter search methods

Normally the problem of finding material parameters cannot be solved implicitly and some kind of 'optimization strategy' has to be used to find material properties yielding acceptable results. An error function is defined to control the optimization.

Often inverse problems do not have a unique solution and the error function may have several local minima in the parameter space.

The most commonly used methods to find optimum parameters, p_α, are based on the error function gradient with respect to the parameters searched for. The derivatives are numerically evaluated by perturbing the parameters one by one.

When all derivatives are available they are used to determine the parameter update. In [1] the parameter update, δ, is determined by

$$(J^T J + \Lambda I)\delta = -J^T \varepsilon \qquad \text{[Eq. 1]}$$

with the Jacobian, J, defined by

$$J = \frac{\partial \varepsilon}{\partial p_\alpha} \ . \qquad \text{[Eq. 2]}$$

The scalar parameter Λ is the Levenberg-Marquardt parameter, controlling the direction and size of the parameter update δ. I is the identity matrix of appropriate size.

This approach is very efficient in a parallel computation environment since the gradients can be evaluated in parallel. If, on the other hand, the derivatives are evaluated sequentially a substantial execution time may be required prior to each parameter update.

The alternative is to update each parameter based on the error function derivative with respect to that parameter only. There is certainly a higher risk associated with using this approach, but the parameters are updated more frequently.

Gradient search methods, of which one is described above, are best suited for situations where the error surface is concave. This is unfortunately not typical for an error function based on vibration shape correlation, with material stiffness as a parameter. Rather, the error surface may have several local minima, where the gradient based methods may fail. An illustration of this is shown in Figure 1, where an error surface with a local minimum (indicated with a red circle) is present. If the initial parameter settings are within that circle, gradient methods are likely to be trapped in the area of the local minimum and consequently the global minima will not be found.

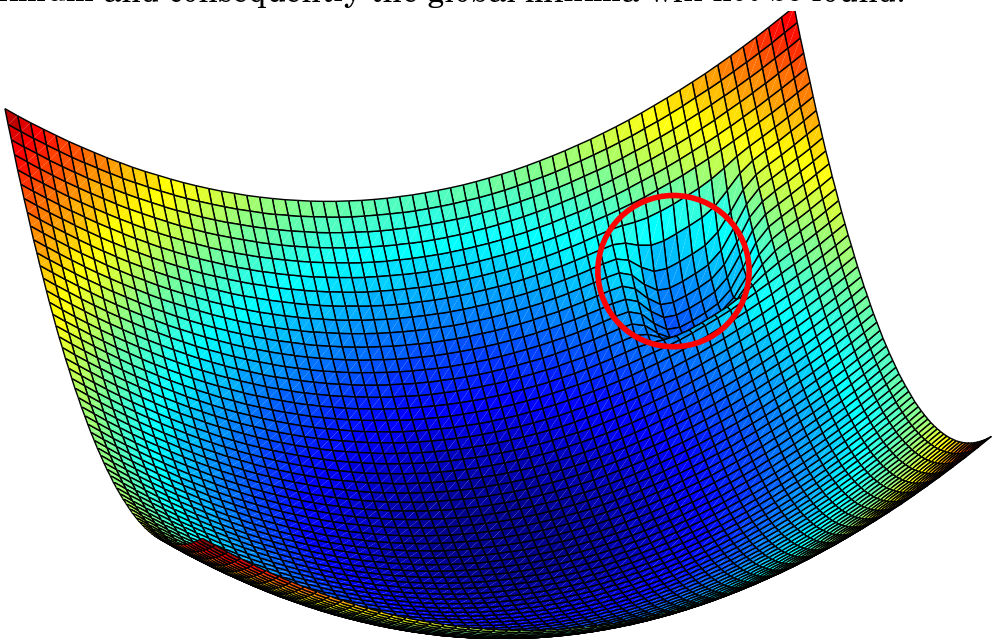


Figure 1: Error surface with a local minima.

2.2 Simulated annealing

For optimization problems with several local minima a concept called ‘simulated annealing’ is sometimes used. With simulated annealing the parameters are initially given large perturbations, i.e the search on the error surface is made with large steps, while at the end of the annealing the steps are small and the aim is to converge to the nearest local minimum.

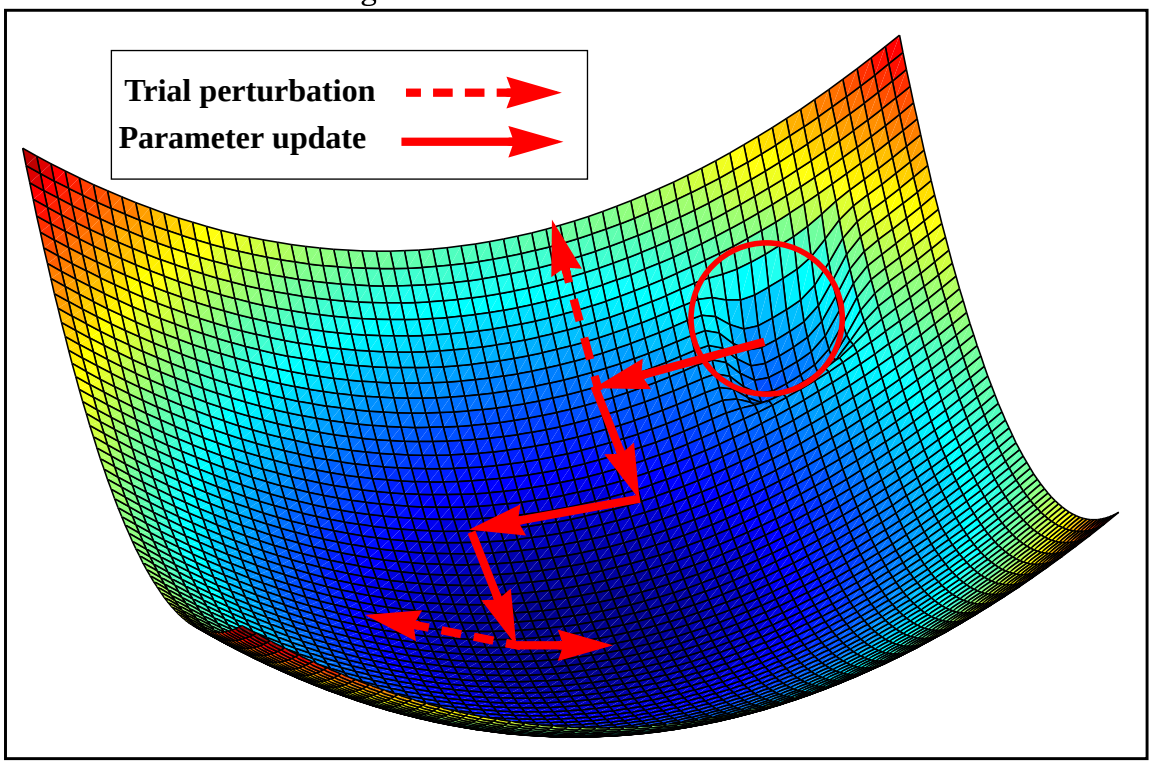


Figure 2: Escape from a local minimum by using large parameter perturbations in ‘simulated annealing’.

The term ‘annealing’ comes from the similarities with an annealing process, where in the beginning the material is very ‘active’ and erupting, but as the temperature decreases the material finally comes to rest. It is common, using simulated annealing, to have a similar decreasing ‘temperature’ variable to control the magnitude of the parameter perturbations.

2.3 Selected parameter search method

A non-concave error surface, with several local minima, is expected in the current application, and with the arguments given above a parameter search method based on simulated annealing was adopted.

An iteration step is taken by modifying one parameter at a time and evaluating the error function prior to and after the modification of this, *active* , parameter. If the value of the error function [Eq. 13] is reduced (i.e., the new parameter setting is better than the current) the active parameter is updated and the increment factor for the active parameter is kept. If, on the other hand, the value of the error function is increased it is likely the search step was performed in the wrong direction, and the update should be to the other side of the current value for the active parameter.

The ‘temperature’ used for decreasing the parameter modification factor is individual for each parameter and is reduced each time the direction of search is changed. The factors used to determine the parameter steps, ζ, are given by

$$\begin{aligned} \zeta_\alpha(k+1) &= \zeta_\alpha^{-\mu}(k) && \varepsilon((k+1) > \varepsilon(k)) \\ \zeta_\alpha(k+1) &= \zeta_\alpha(k) && \varepsilon((k+1) \le \varepsilon(k)) \end{aligned} \qquad \mu < 1 \ . \qquad \text{[Eq. 3]}$$

Figure 3 illustrates the case of a trial parameter step improving the correlation (upper) and the case when the search direction is changed (lower).

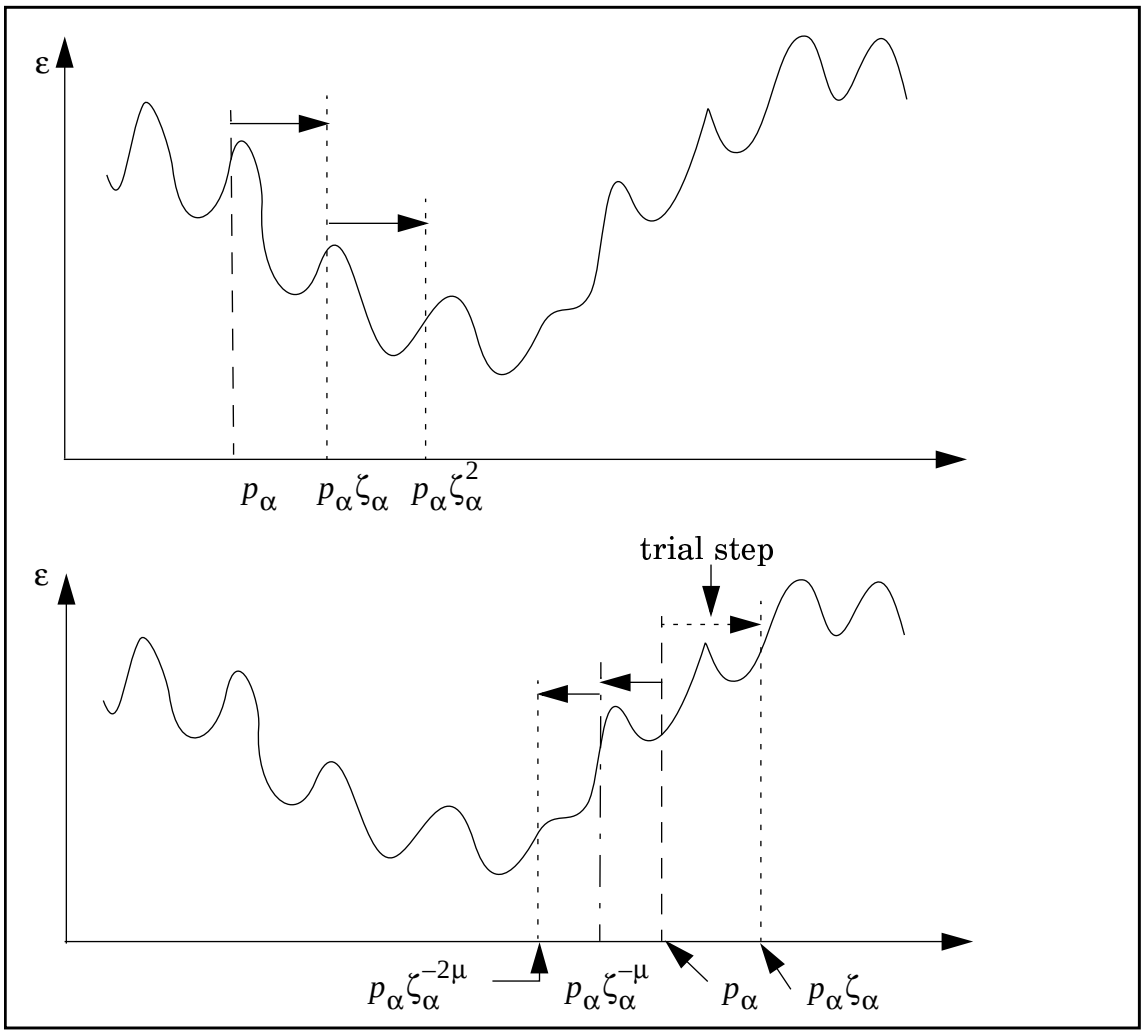


Figure 3: The parameter search method for the one dimensional case. Upper: The first step is taken in the ‘right’ direction. Lower: The first step is taken in the ‘wrong’ direction

This means that if the search is performed in a direction giving an increased value for the error function the update is to a value on the other side of the initial value of the design variable. The step-size is shortened each time the search direction is changed, ensuring convergence.

3 *Application example*

The Saab 340 acoustic test section (Figure 4) was selected as an application example.

Figure 4: The Saab 340 acoustic test section.

In the BRITE/EURAM project, ASANCA II, a very extensive ground vibration test, involving the Saab 340 acoustic test section, was performed [2]. A large database of test results, suitable for inverse modeling, is available from these tests [3].

It is also interesting to compare the results from inverse modeling with modeling results obtained in the ASANCA II project, where 'U-vector Expansion modeling' (UEM) and finite element modeling of the Saab 340 acoustic test section was performed [4], [5].

3.1 Test data preparation

The fuselage vibrations in the radial direction is available for 140 locations on the outside of the skin. Structural excitation, by means of attached electrodynamic inertia shakers, was applied at 108 of these locations. Sinusoidal excitation signals at 30 frequencies in the region 60-130 Hz were used. The primary interest was to determine the dynamic properties at the Saab 340 Blade Passage Frequency (BPF) which is 82 Hz at normal flight conditions. Consequently the inverse modeling was based on the 82.0 Hz responses, but some results for other frequencies will also be presented here.

3.2 Finite element model (general)

The basic idea was to define a geometrically very simple model of the fuselage, one not representing structural components individually.

Simply looking at the Saab 340 structure it is easy to realize that the structure has to be divided into at least three parts:

- *cabin wall above floor level (yellow in Figure 5)*
- *cabin floor (red in Figure 5)*
- *cabin wall below floor level (blue in Figure 5)*

Initially this was thought to be only a preliminary approximation, but as can be seen later in this report, it appears that a model with only three different materials, each assigned to one of the regions defined above, is capable of representing the measured dynamic properties of the fuselage section fairly well.

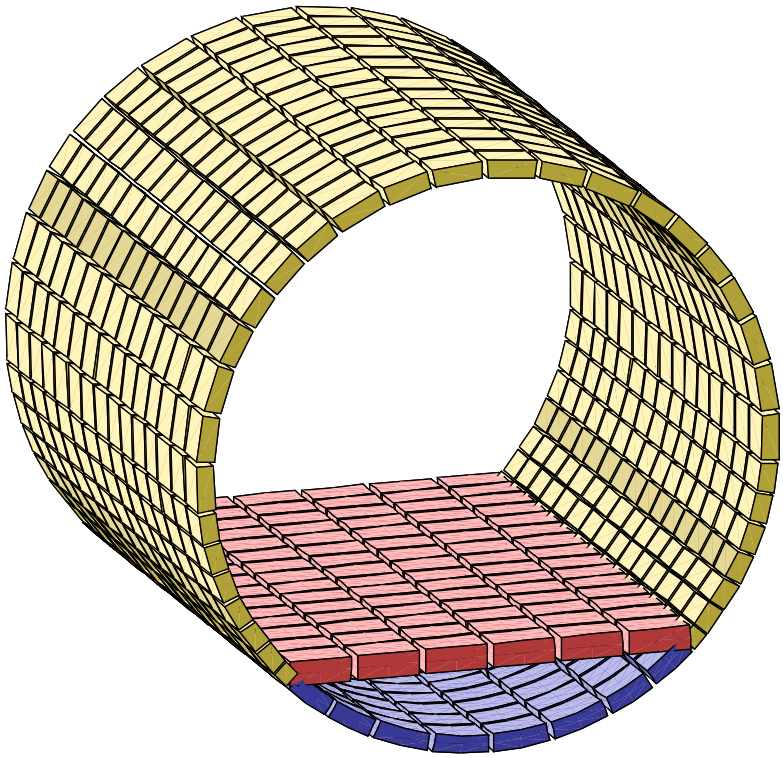


Figure 5: The finite element model with the three regions identified by color.

It is obvious that the material model to be used for the conglomerate of fuselage skin, frames, stringers, interior panels and thermal insulation has to be quite general.

The elements used are modified 8-node solid elements. This selection of elements is motivated in section 3.4, where the element is described.

The complete finite element model consists of 720 elements and 4512 degrees of freedom (Figure 5).

3.3 Material properties

It was assumed that different material properties would be necessary for the fuselage structure above and below the floor level respectively, and for the floor. With the arguments given above, orthotropic material models are assumed for the three parts of the model. The principal directions, required for an orthotropic material, are defined by Figure 6.

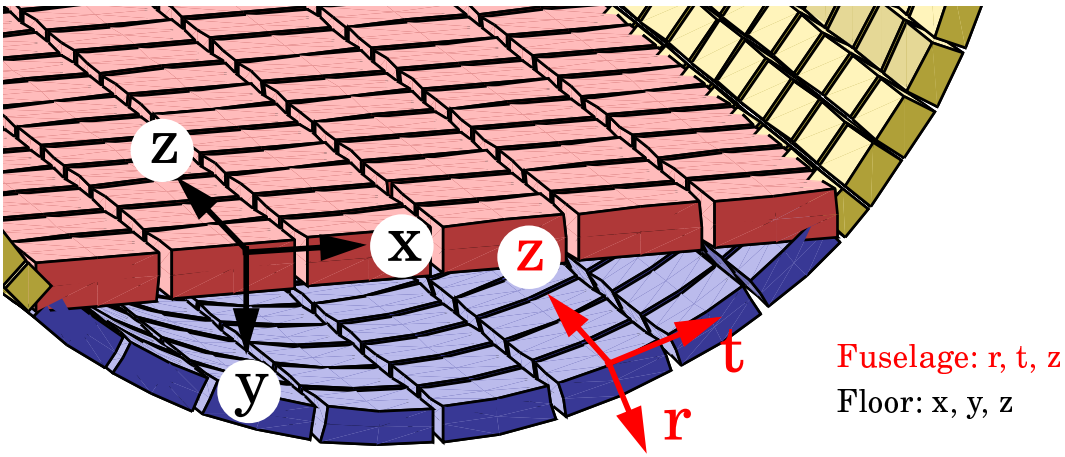


Figure 6: Definition of directions.
Fuselage: Radial (r), Tangential (t) and Longitudinal (z).
Floor: Horizontal (x), Vertical (y) and Longitudinal (z).

In order to write the constitutive equations for all three materials in the same form, three general directions, called '1', '2' and '3', are introduced according to table 1.

TABLE 1. Material direction definitions.

Model part	direction '1'	direction '2'	direction '3'
Fuselage	Radial (r)	Tangential (t)	Longitudinal (z)
floor	Horizontal (x)	Vertical (y)	Longitudinal (z)

A central property of the present modeling is the demarcation to a very limited frequency range. This simplifies the selection of material model since the frequency dependencies of the material properties are not required. A further simplification is introduced by the use of a stiffness proportional damping. Although this seems to be a very crude way of handling the damping properties, it can be shown (e.g., [6]) that stiffness proportional damping is equivalent to more advanced damping models like the ones based on Augmenting Thermodynamic Fields [7].

The constitutive relation may be written in matrix form

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & & & \\ D_{21} & D_{22} & D_{23} & & & \\ D_{31} & D_{32} & D_{33} & & & \\ & & & D_{44} & & \\ & & & & D_{55} & \\ & & & & & D_{66}\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{12}\\ \varepsilon_{13}\\ \varepsilon_{23}\end{bmatrix} \qquad \text{[Eq. 4]}$$

where the D_{ij} are the complex valued material properties to be found with the inverse modeling technique. Orthotropic symmetry is assumed (D_{ij}=D_{ji}). Each term D_{ij} may be written

$$D_{ij} = D_{ij}^{\Re e}(1 + i\xi_{ij}) \qquad \text{[Eq. 5]}$$

where $D_{ij}^{\Re e}$ is the elastic part and ξ_{ij} determines the individual damping related to each component of the stress-strain relation. As a first approximation all ξ_{ij} are given the same value, but this approximation may be reconsidered if the results of the inverse modeling are not sufficient. However, without this assumption 18 material constants would be required for each material instead of 10.

3.4 Element selection

A very simple, and consequently numerically efficient, element was desired. However, the element formulation should give acceptable results for the type of deformation found in tests with bending dominant behavior of the fuselage structure.

After some literature review an ‘enhanced strain formulation’ 8-node solid element [8] was selected. It may seem to be an awkward selection, using geometrically linear solid elements for a curved rather thin structure with a dominant bending behavior. However, the additional deformation modes, introduced with the ‘enhanced strain formulation’, means that the element works sufficiently well in bending [9].

The results of the inverse modeling will show if the element selection is suitable to model the present structure.

During the test the fuselage structure was supported by six soft, inflated, tire-tubes. These supports are simulated with 3-axial springs. The stiffnesses of the springs are assumed equal for all three directions (and the six individual supports) and this stiffness is a separate parameter in the inverse modeling.

3.5 Dynamic equilibrium

With the material model selected in section 3.3 we can write the dynamic equilibrium equation in a very simple form

$$M\ddot{x}(t) + Kx(t) = f(t) \quad . \qquad \text{[Eq. 6]}$$

For a harmonic excitation, given by

$$f(t, \omega) = Fe^{i\omega t} \qquad \text{[Eq. 7]}$$

where F is a (possibly complex-valued) constant, the steady-state response, provided the system is linear, is given by

$$x(t) = x(f(t, \omega)) = Xe^{i\omega t} \qquad \text{[Eq. 8]}$$

where X is a complex-valued constant. Using this expression for $x(t)$ and evaluating the time derivatives of $x(t)$, [Eq. 6] can be written

$$(-\omega^2 M + K)Xe^{i\omega t} = Fe^{i\omega t} \quad . \qquad \text{[Eq. 9]}$$

The frequency response functions for a system at frequency ω are defined by

$$H_{ij}(\omega) = x_i / f_j = X_i / F_j \qquad \text{[Eq. 10]}$$

where x_i is the response at dof i for a force f_j at dof j. By simply applying a unit force at each driving point used in the test, the model estimate of the frequency response functions H_{ij} are found as the relation X_i / F_j from [Eq. 10].

3.6 Error function

The error function selected is the correlation between measured response, X_t, and the model response, X_a, for a particular excitation. Two quantities are defined to determine the correlation; the vibration shape and the phase correlation, λ, defined by

$$\lambda = \frac{\Re e(X_t^H X_a)}{|X_t||X_a|} \qquad \text{[Eq. 11]}$$

and the average response level correlation, Δ, defined by

$$\Delta = \frac{min(|X_t|, |X_a|)}{max(|X_t|, |X_a|)} \quad . \qquad \text{[Eq. 12]}$$

The first parameter (λ) is somewhat similar to the ordinary coherence function for two vectors. For real-valued vectors, λ will become the square root of the coherence function. The reason for not using the ordinary coherence function is that it is not influenced by phase rotations and will consequently not be a good correlation indicator for the complex-valued response vectors to be compared in the present application.

Deviations in vibration shape and phase are considered to be more severe than deviation in response amplitude. This is encountered by giving λ a higher influence on the error function than the amplitude correlation (Δ).

$$\varepsilon = (1-\lambda^{\kappa}\Delta^{1/\kappa}) \qquad \kappa \geq 1 \qquad \text{[Eq. 13]}$$

In present application $\kappa=2$ is used.

3.7 Initial parameter settings

The 31 parameters to be found are numbered 1-31 according to *TABLE 2*.

TABLE 2. Parameter identification. Parameters belonging to material 2 and 3 given in parenthesis.

Parameter	1	2 (3,4)	5-10 (14-19, 23-28)	11-13 (20-22, 29-31)
Variable	Support springs	Damping (ξ)	$D_{ii,\ i=1:6}$	$D_{ij,\ i=1,2,3\ j=1,2,3\ i\neq j}$

The selection of starting values was based on a few trial and error test, trying to get somewhat close in response level. *TABLE 3* gives the starting value for the material properties.

TABLE 3. Initial material parameters

Fuselage	Damping ζ $K=\overline{K}(1+i\zeta)$	$D_{ii,\ i=1,2,3}$	$D_{ii\ ,\ i=4,\ 5,\ 6}$	$D_{ij,\ i=1,\ 2,\ 3\ j=1,2,3}$
Top	0.5	100 MPa	50 MPa	5 MPa
Bottom	0.5	100 MPa	50 MPa	5 MPa
Floor	0.5	100 MPa	50 MPa	5 MPa

For the support springs, simulating the inflated tire-tubes on which the fuselage section was placed during the test, the initial stiffness was set to $1*10^5$ N/m.

The scaling factors to be used for the parameter update should be sufficiently large to scan a large area of the error surface, although small enough to permit convergence in reasonable time.

Some trial runs were performed to investigate the influence of the different parameters, leading to the initial scaling factor selection given in *TABLE 4*.

TABLE 4. Initial scaling factors.

Parameter	1	2-5, 7-31	6 *
Scaling factor	10	5	2

* (D_{tt} for fuselage section above floor level)

3.8 Parameter search strategy and perturbations

Not surprisingly, some parameters are more important than others. This is, for example, the reason for the lower scaling factor selected for parameter 6 (*TABLE 4*). Further, it appears wise to start by finding reasonable values for the parameters having the largest influence on the results and then continuing the search involving all parameters.

Three parameters (D_{tt},D_{tz} and the damping for the fuselage section above the floor level) were identified as the most important parameters. Five iterations were performed initially involving only these three parameters.

In all simulations the convergence factor, μ, was 0.5 (see [Eq. 3]).

4 Inverse modeling results

4.1 Results presentation

The inverse modeling is based on the frequency response functions measured at the Blade Passage Frequency (82.0 Hz). For each modeling case, using different parts of the available experimental data, the following checks were used:

- *Convergence history for error function, shape/phase correlation and amplitude correlation*
- *Obtained material parameters*
- *Individual correlation for the 112 columns in the frequency response function matrix (shape/phase and amplitude)*
- *Response shape comparison for one driving point (point id: '363R10')*

In addition frequency sweep results for the measurement range 60-130 Hz are used for:

- *Average correlation for all 112 frequency response function columns at each frequency*
- *Driving point receptance (displacement /force) for driving point id: '363R10'*

The response shape comparisons are made using a model defined by the measurement response locations. For this the finite element analysis results are transformed to pure radial vibrations at the vibration measurement locations.

In addition, the responses of the complete finite element models are visualized in Appendix A.

4.2 Case 1: Using all 112 columns of the frequency response function matrix.

The five initial iterations, involving only three parameters, brings the results to the dashed line in Figure 7. Subsequent iterations, now involving all 31 parameters, mainly improve the correlation in response shape and phase. After approximately 200 iteration the solution has essentially converged. An average correlation in vibration shape/phase of 0.86, and an amplitude correlation of 0.75 was achieved after the 325 iterations.

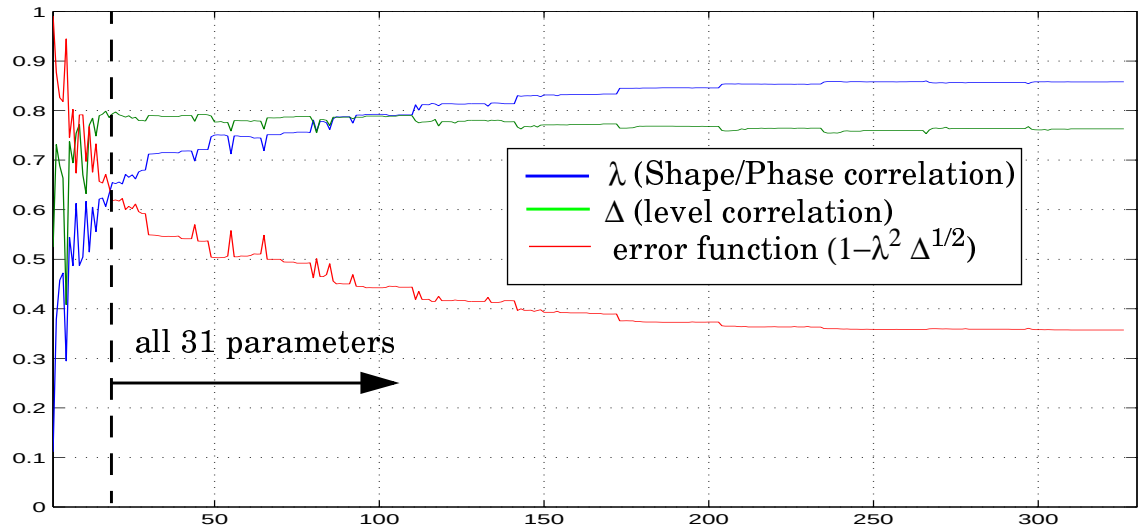


Figure 7: Convergence history (all 112 columns).

The correlation for each of the 112 frequency response function columns, each related to a specific driving point (except the four excitation points used twice), is illustrated in Figure 8.

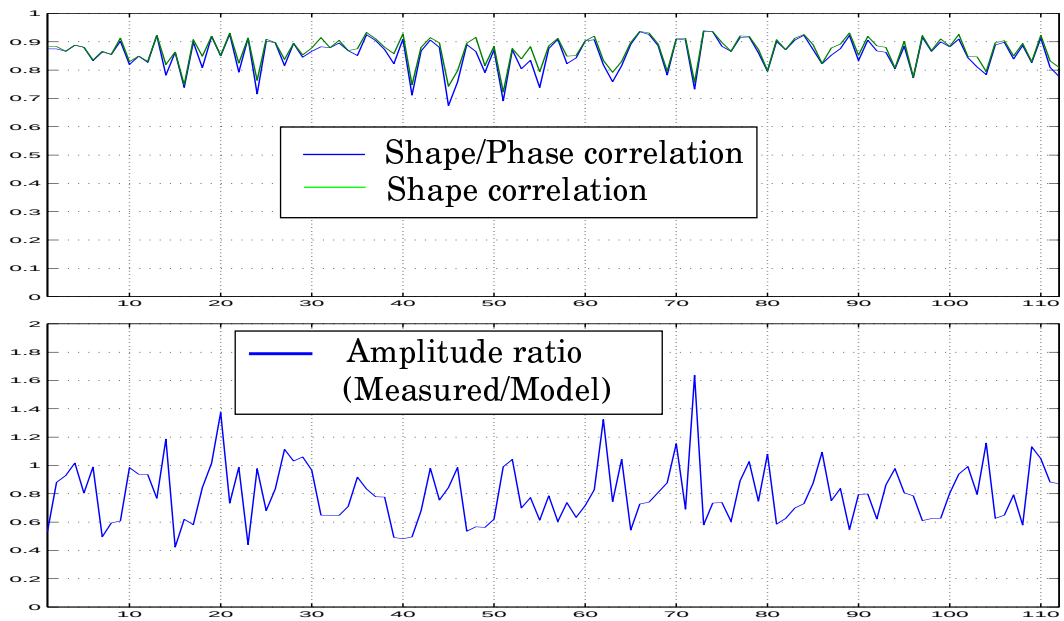


Figure 8: Correlation for individual frequency response function columns.

A separation of vibration *shape* correlation and vibration *phase* correlation is of interest in some cases. When using models to design Active Noise and Vibration Control, ANVC, a systematic phase difference can be compensated for by the phase of the actuator driving signals. However, only some of the columns (driving points) show identifiable difference between the pure shape correlation and the shape/phase correlation (Figure 8).

The material parameters found at the end of the parameter search are given in Table 5.

TABLE 5. Material parameters

	Material		
Parameter	**Fuselage Above floor**	**Floor**	**Fuselage Below floor**
Damping ζ	0.129	3.9312	0.0109
$\mathbf{D_{11}}$	676 MPa	399 MPa	1.67 GPa
$\mathbf{D_{22}}$	317 MPa	2336 GPa	2.18 GPa
$\mathbf{D_{33}}$	5.59 GPa	$9.8*10^5$ GPa	$9.8*10^5$ GPa
$\mathbf{D_{44}}$	59.6 MPa	618 MPa	836 MPa
$\mathbf{D_{55}}$	4.95 MPa	9.04 MPa	250 MPa
$\mathbf{D_{66}}$	226 MPa	2.80 GPa	12.2 MPa
$\mathbf{D_{12}}$	5.53 MPa	2.24 MPa	5.35 kPa
$\mathbf{D_{13}}$	7.48 MPa	0.018 MPa	21.5 MPa
$\mathbf{D_{23}}$	0.366 MPa	0.200 MPa	279 MPa

Some of the values in Table 5 are quite unrealistic, given the actual materials and dimensions of the fuselage relative to the dimensions of the assumed dimensions of the model. This is a result of coupling between the influence of different D-matrix settings. Basically a modification of one term in the D-matrix can be compensated for by changes of other terms in the D-matrix. In mathematical terms the system is under-determined.

In most situations it may be possible to specify some of the D-matrix parameters, or at least a range for them. This will make the inverse problem easier and will lead to more realistic material parameters. However, the models derived from under-determined inverse modeling may still be accurate and of great practical use even though the material parameters are not realistic.

4.2.1 Vibration shape correlation

Averaged correlation values are suitable for error functions and for comparing different modeling methods. However, deformation shapes are required to get a real impression of the actual correlation between measured and computed responses. A typical example of this is showed below (Figure 9). The selection of excitation point was made based on previous work using the U-vector Expansion Method (UEM) [4].

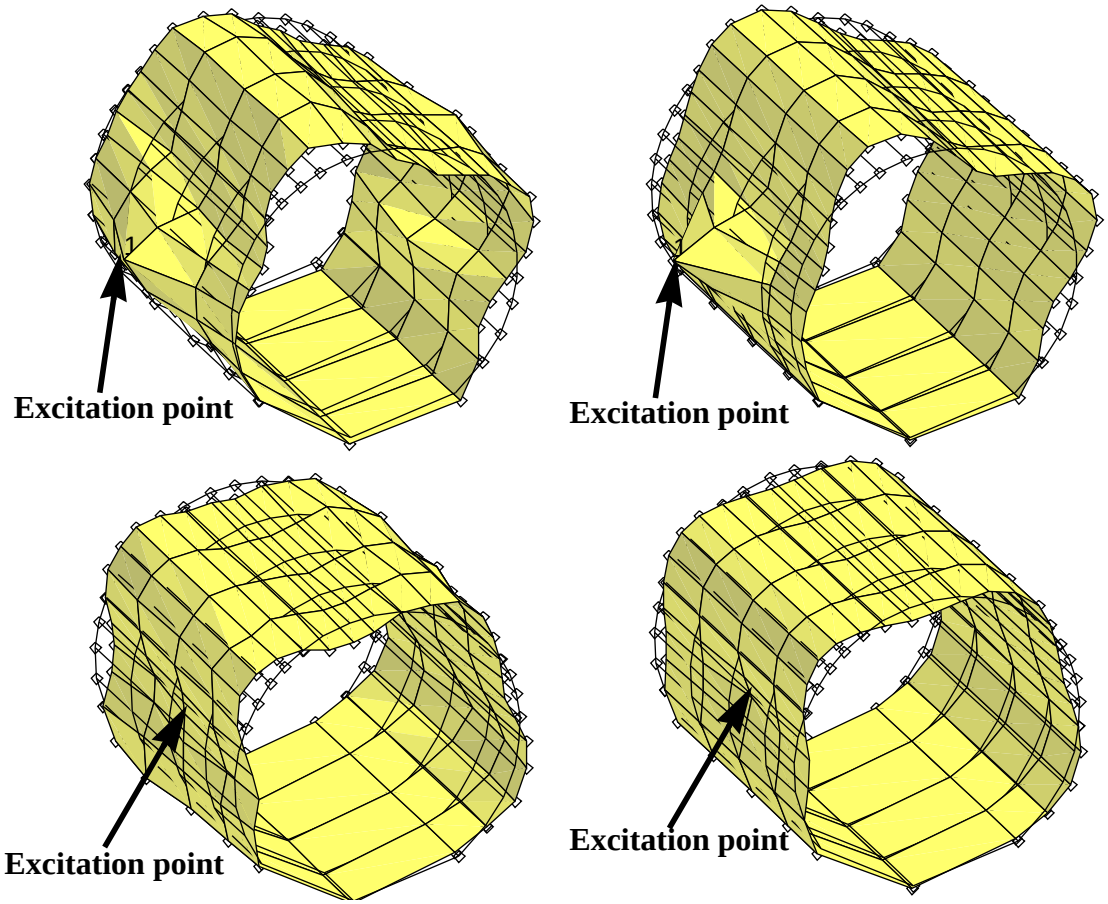


Figure 9: Frequency response function response for excitation at point id: '363R10' (82.0 Hz).
Left: Measured, Right: The finite element model.
Upper: Real part, Lower: Imaginary part (force: 1+0i).

The deformation correlation for the response in Figure 9 is slightly higher than the average correlation (0.908 vs. 0.86), although 19 of the columns are showing even higher correlation. Concerning the level correlation, 34 of the columns are better than the one selected for Figure 9.

4.2.2 Frequency band results

Although the major interest is focused on 82 Hz (the Saab 340 Blade Passage Frequency), some comparisons of the frequency response functions at other frequencies are interesting. A frequency sweep was performed, using the finite element model with the material constants determined at 82.0 Hz, over the measurement frequency range 60-132Hz (Figure 10).

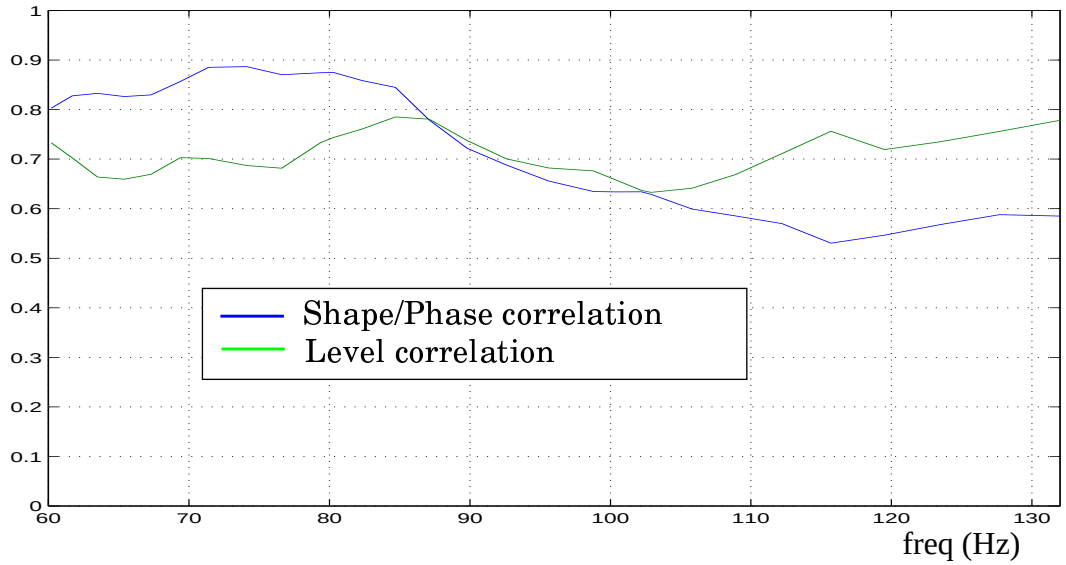


Figure 10: Correlation over the frequency range 60-130 Hz.

The correlation over the frequency range (60-132Hz) is quite good, although a degradation with increasing frequency is apparent.

Another interesting comparison is the driving point response over the frequency range. Again the driving point called '363R10' is used to show an example of how the driving point *receptance* (displacement over force) varies with frequency (Figure 11).

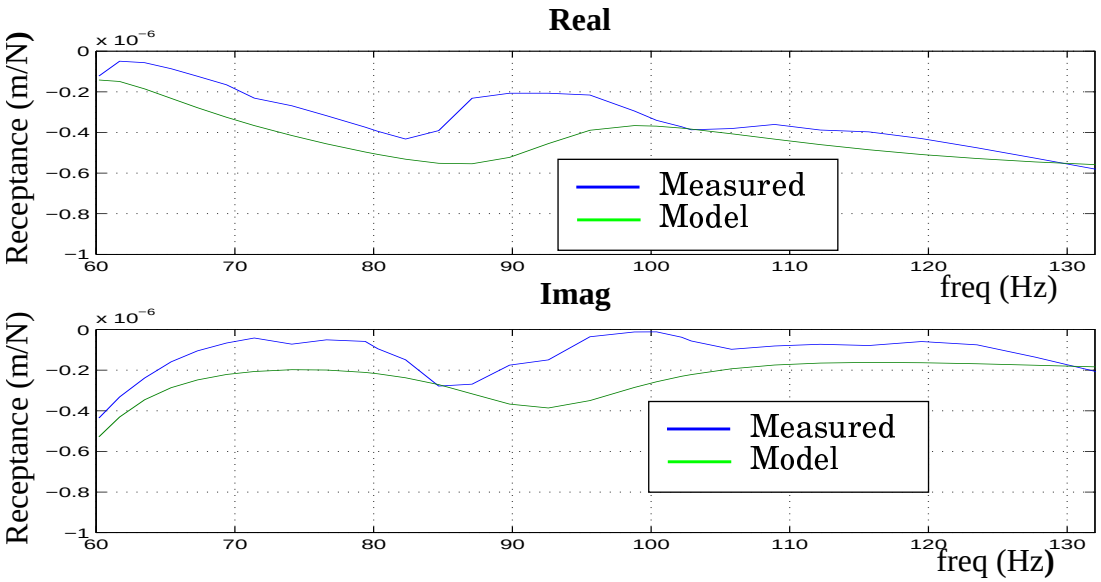


Figure 11: Driving point receptance for point id: '363R10'.

Also this comparison indicate interesting model properties, giving good result over the whole frequency range 60-130 Hz.

4.3 Case II: Using 12 of the columns of the frequency response function matrix.

For this inverse modeling, only small sub-sets of the data were used, giving a more representative simulation of a practical situation where only a few driving points are used (and possibly fewer response locations). The same initial values, scaling factors and the initial run involving only a few parameters was used.

4.3.1 Selection of test data

A random selection of 12 driving points to be included was made (simply every tenth column in the frequency response function matrix).

4.3.2 Results

The convergence is similar to that for the case using all (82.0 Hz) test data, although this time only the 12 selected columns were involved in the correlation.

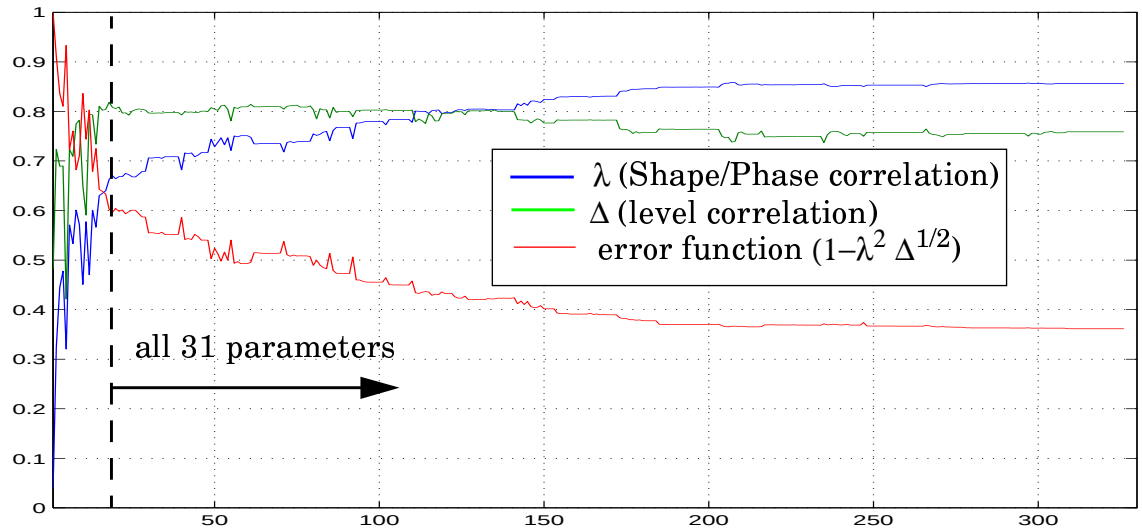


Figure 12: Convergence history (for the selected 12 columns).

A natural effect of disregarding some of the test data would be similar or better results for the part of the data included in the inverse modeling, while the correlation between measured and model-derived frequency response functions not included in the inverse modeling should decline. Surprisingly this effect was very small, which can be seen in Figure 13, where the correlation for all the 112 driving points (columns) are shown. Actually the average correlation in shape/phase for the 12 frequency response function columns used for the inverse modeling is only about 2% better than for the 100 columns not used (0.86 vs. 0.84). For the amplitude correlation the difference is also only marginal (0.77 vs. 0.76).

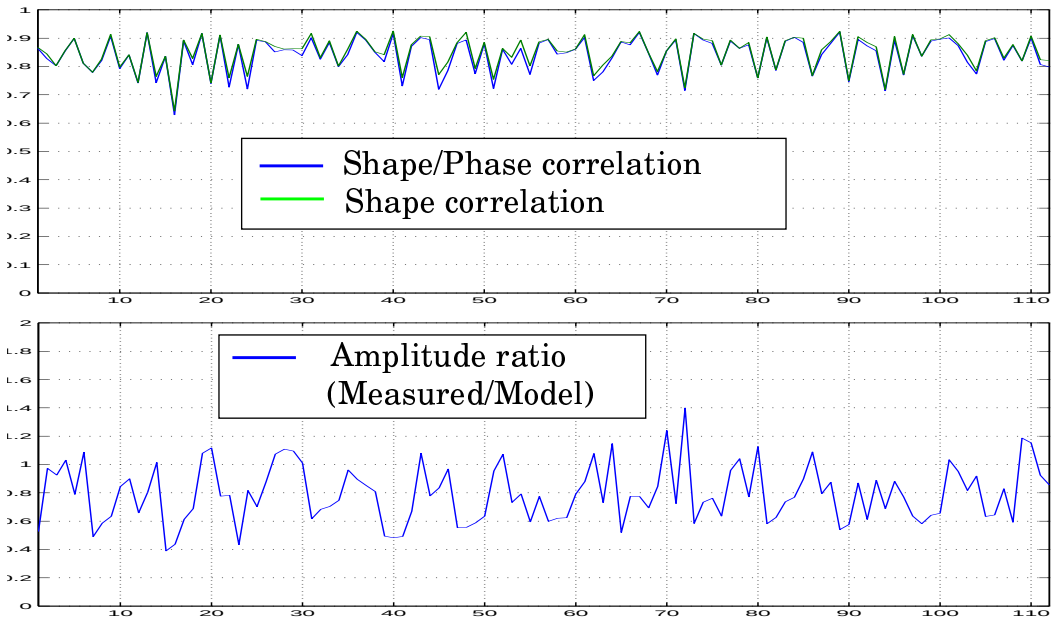


Figure 13: Correlation for individual frequency response function columns.

The material parameters after 325 iterations (*TABLE 6*) are almost completely different from the values given by the inverse modeling involving data for all driving points (*TABLE 5*).This confirms the expected under-determined structure of the inverse modeling problem.

TABLE 6. Material parameters (12 driving points)

	Material		
Parameter	**Fuselage Above floor**	**Floor**	**Fuselage Below floor**
Damping ζ	0.161	1.367	0.409
D_{11}	9.17 MPa	1.24 GPa	38.5 MPa
D_{22}	376 MPa	0.654 MPa	56.1 MPa
D_{33}	1.67 GPa	1.67 GPa	25.3 GPa
D_{44}	29.7 MPa	5.11 GPa	4.95 MPa
D_{55}	4.25 MPa	1.09 MPa	35.2 MPa
D_{66}	263 MPa	22.4 MPa	74.8 MPa
D_{12}	5.53 MPa	1.6 kPa	5.53 MPa
D_{13}	7.48 MPa	33.8 MPa	2.24 MPa
D_{23}	1.22 MPa	2.24 MPa	3.02 MPa

4.3.3 Vibration shape correlation

The response for the previously used driving point '363R10' is shown in Figure 14. This is one of the driving points assumed not included in the test data. The correlation for this driving point is 0.85 for shape/phase and 0.81 for the response amplitude.

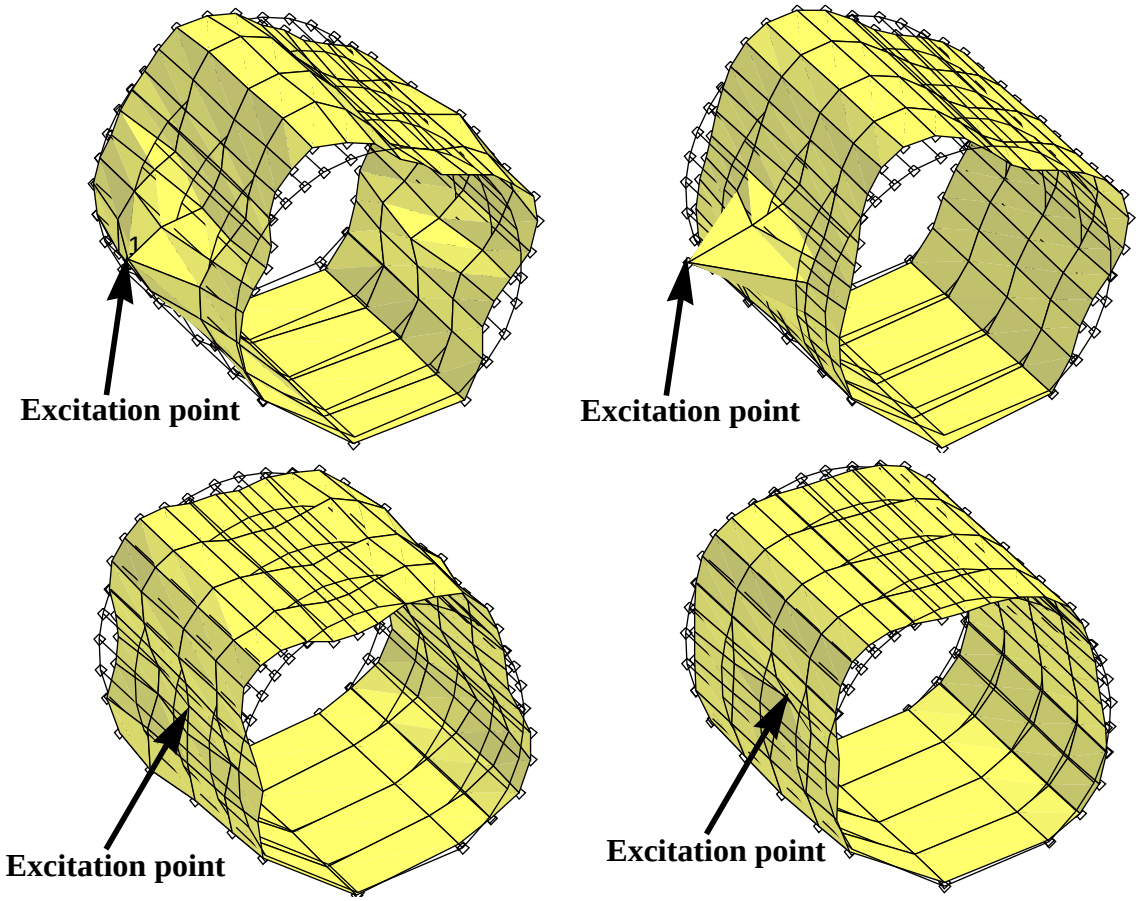


Figure 14: Frequency response function response for excitation at point id: '363R10' (82.0 Hz).
Left: Measured, Right: The finite element model.
Upper: Real part, Lower: Imaginary part (force: 1+0i).

The lower correlation, compared to the Case I results, can be seen in the area of the driving point, but in general the measured vibration shape agrees quite well with the finite element model results.

4.3.4 Frequency band results

Despite the large differences in material properties compared to the Case I results, the correlation over the 60-130 frequency range is very similar (Figure 15).

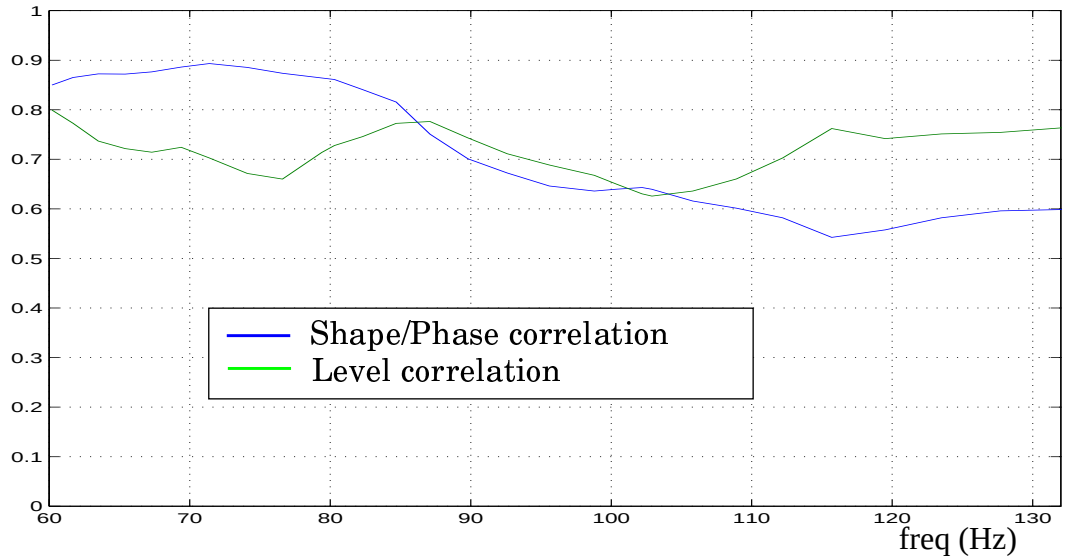


Figure 15: Correlation over the frequency range 60-130 Hz.

The driving point receptance (for driving point id: '363R10'), however, is not as good as the result for Case I (Figure 16).

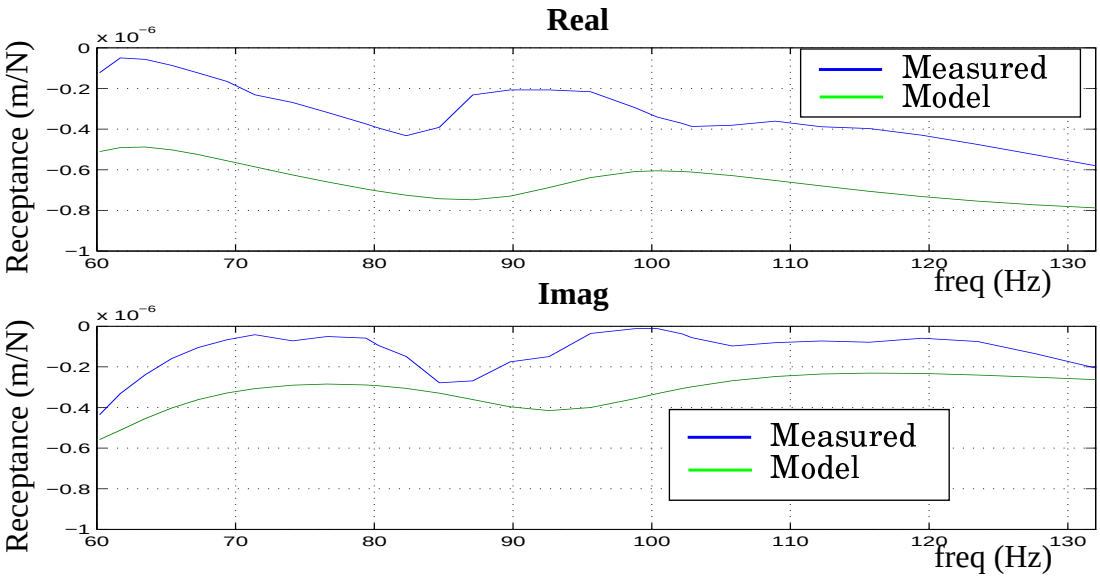


Figure 16: Driving point receptance for point id: '363R10'.

The major difference is in the magnitude of the receptance, while the frequency dependence is rather similar.

4.4 Case III: Using 12 of the columns of the frequency response function matrix.

Inspired by the results when using data for 12 driving points, a simulation using even fewer driving points (6) was performed.

4.4.1 Selection of test data

One driving point was selected on each of the five test section fuselage frames. Further, the driving points were spread out over the circumference of the fuselage, aiming to excite different vibration shapes of the structure.

4.4.2 Results

Again no major differences are found for the convergence history compared to the case of using data for all driving points in the inverse modeling (Figure 17).

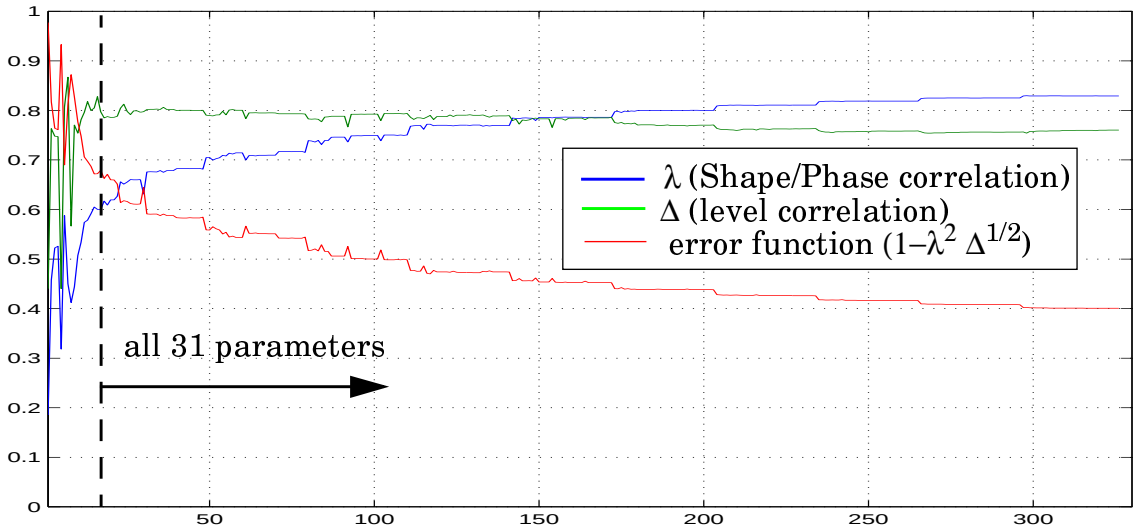


Figure 17: Convergence history (for the 6 selected columns).

The slightly lower correlation than for the two previous cases may indicate that the measurement data for the six selected driving points are less accurate than average, or that the responses for these driving points are more difficult to model than the response for most other driving points.

Also, the correlation for the 106 columns not included in the present inverse modeling is lower (shape: 0.80 vs. 0.86, amplitude: 0.70 vs. 0.77). The individual correlations are shown in Figure 18, and in this figure it can be seen that there is a difference, although small, between the shape correlation and the combined shape/phase correlation.

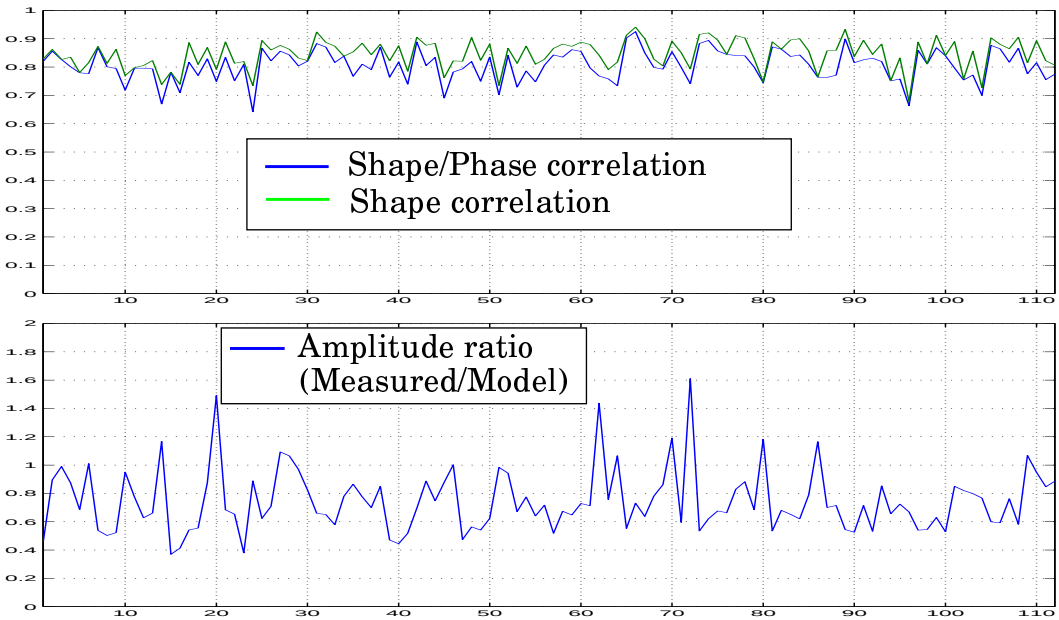


Figure 18: Correlation for individual frequency response function columns.

The resulting material parameters after the 325 iterations are somewhat similar to the results for Case II (using 12 columns), although some significant differences can be found (*TABLE 7*).

TABLE 7. Material parameters (6 driving points)

	Material		
Parameter	**Fuselage Above floor**	**Floor**	**Fuselage Below floor**
Damping ζ	0.150	0.192	0.250
$\mathbf{D_{11}}$	13.4 MPa	1.67 GPa	18.1 MPa
$\mathbf{D_{22}}$	385 MPa	0.358 MPa	41.5 MPa
$\mathbf{D_{33}}$	3.06 GPa	0.654 MPa	370 MPa
$\mathbf{D_{44}}$	26.0 MPa	1.99 GPa	6.69 MPa
$\mathbf{D_{55}}$	4.95 MPa	19.2 MPa	26.0 MPa
$\mathbf{D_{66}}$	253 MPa	1.48 MPa	19.2 MPa
$\mathbf{D_{12}}$	5.53 MPa	5.35 kPa	5.96 MPa
$\mathbf{D_{13}}$	7.48 MPa	11.8 MPa	10.1 MPa
$\mathbf{D_{23}}$	1.22 MPa	1.92 MPa	2.24 MPa

4.4.3 Vibration shape correlation

Surprisingly enough, the response for driving point '363R10' tends to be slightly better than in the previous section (using 12 driving points). The correlation is similar, 0.85 for shape/phase and 0.84 for level (Case II: 0.85 and 0.81 respectively).

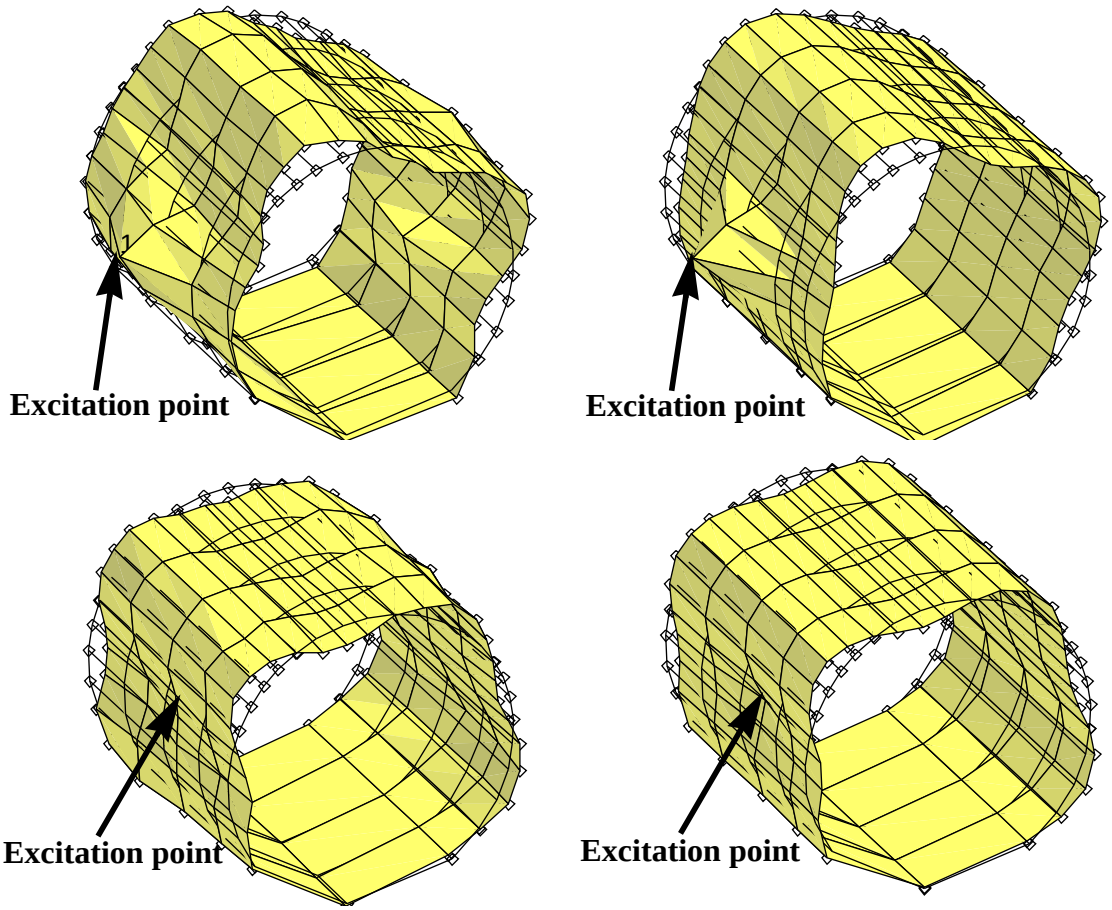


Figure 19: Frequency response function response for excitation at id: '363R10' (82.0 Hz).
Left: Measured, Right: The finite element model.
Upper: Real part, Lower: Imaginary part (force: 1+0i).

However this is only one example of response shape correlation, and the average correlation (see Section 4.4.2) is naturally a better indicator of the model quality.

4.4.4 Frequency band results

Again the frequency sweep gives surprisingly good results (Figure 20).

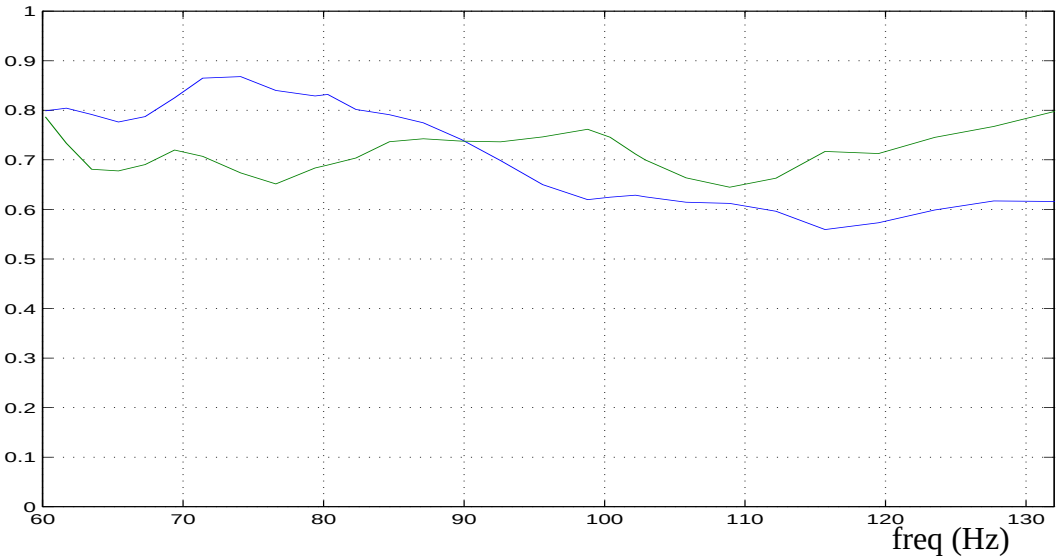


Figure 20: Correlation over the frequency range 60-130 Hz.

For the driving point receptance (point '363R10') the results are also quite good (Figure 21), with quite accurate amplitude over the frequency range 60-130 Hz.

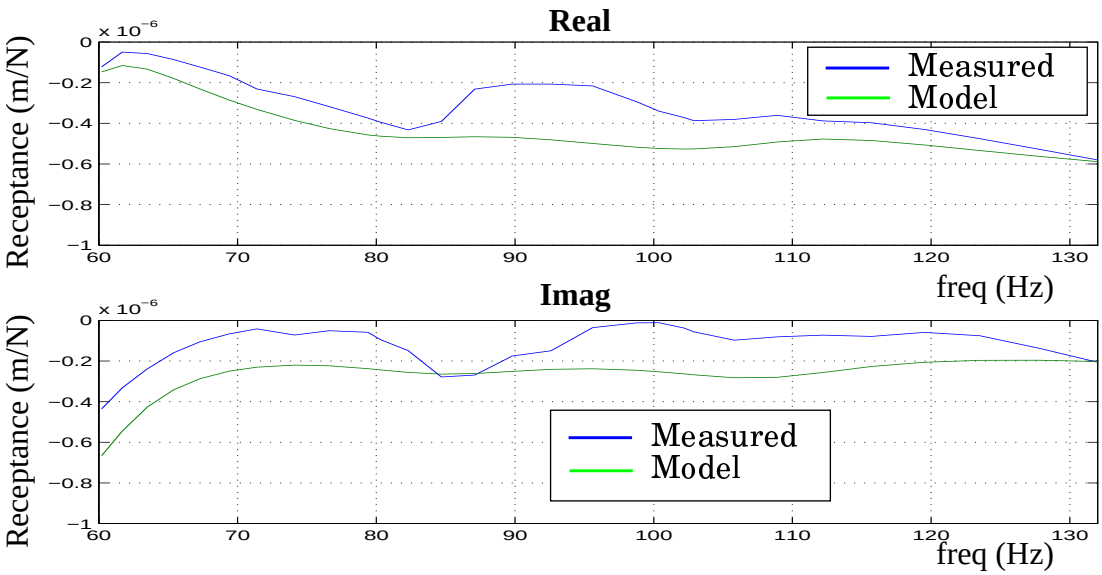


Figure 21: Driving point receptance for point id: '363R10'.

Again, this is just one example, but it gives an impression of the quality of the model.

5 Conclusions

An inverse modeling technique has been used to model the dynamic properties of a turbo-prop aircraft structure. The major objective was to derive models to be used for analysis at the Blade Passage Frequency (BPF) and only experimental measurement data obtained at this frequency was used for the inverse modeling. It was found that the resulting models not only give acceptable results at the BPF, but also in a frequency band 60-130 Hz.

The present inverse modeling problem is under-determined to its character, resulting in models with very different material parameters, but giving similar results. One reason for this is the absence of vibration measurements at the floor, which virtually converts the floor to a boundary specification for the fuselage.

Introducing upper and lower bounds for the parameters in the inverse modeling would also reduce the problem with non-unique solutions.

In the present study the mass distribution was not very refined, basically the total mass of the structure was distributed equal over the simplified geometrical model. Heavy, but geometrically rather small, components like the windows are consequently not represented accurately. In principle the true mass distribution could be used even though the stiffness properties are 'smeared' for large areas of the structure. This could give better results, and if the structure is available as a CAD model the mass distribution would be easily determined.

Despite the quite simple 8-node solid element used, the model is capable of reproducing quite complicated vibration shapes, not only at the frequency for which the material parameters are optimized, but over a rather wide frequency band.

The results achieved are comparable with the results of the U-vector Expansion Modeling results presented in [4]. Comparing the present results with the results of the large finite element model of the Saab 340 developed at FFA [5] is not really appropriate since the concept used by FFA does not require any test data. However, the results of the present inverse modeling study encourage the use of the inverse modeling technique for aircraft structures.

From a computational point of view the present study was far from optimal. All computations were performed using the sparse matrix functions in MATLAB . One iteration was performed in about 300 seconds, leading to a total execution time of 80 hours for the three simulations. Implementation of the concept in a modern parallel computing environment would reduce the computation times significantly. A preliminary study indicates a possible speed-up by a factor 10 or more using optimized, machine specific, program code.

References

[1] **Schnur D. S, Zabaras Nicholas**
An inverse method for determining elastic material properties and a material interface.
Int. Journal for num. methods in engineering, Vol. 33, 2039-2057 (1992).

[2] **Gustavsson M.**
Test Plan for Saab 340 Test Section.
ASANCA II Documentation, Deliverable 64/A, Saab Aircraft AB, 1995.

[3] **Gustavsson M.**
Test Report on Saab 340 Test Section Ground Test.
ASANCA II Documentation, Deliverable 66/1B, Saab Aircraft AB, 1996.

[4] **Gustavsson M.**
Dynamic modeling of a turbo-prop aircraft using the U-vector Expansion Method.
Modern Practice in Stress and Vibration Analysis, Gilchrist (ed.) 1997 Balkema, Rotterdam.

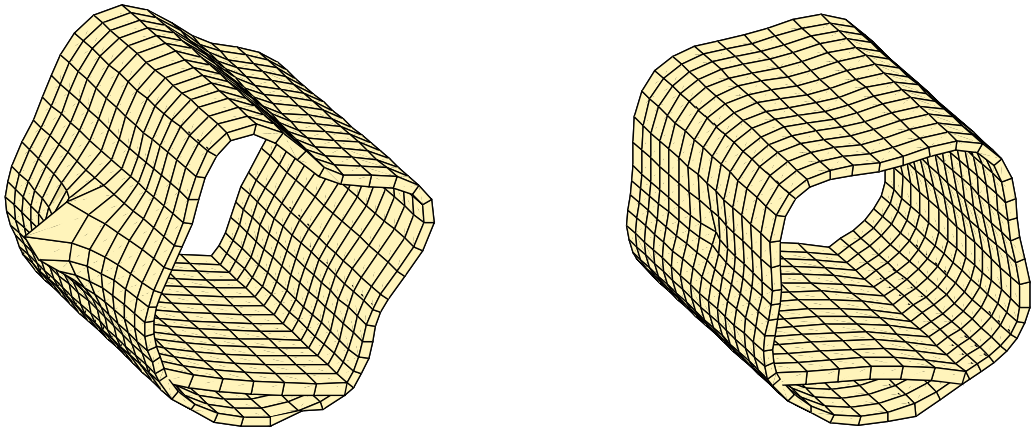
[5] **Hörlin N, Tengzelius U.**
FRF:S from Analysis of Vibro-Acoustic FE Model of Trimmed Saab 340 Test Section
ASANCA II Documentation, Deliverable 66/2a, FFA Stockholm 1997.

[6] **Dovstam K.**
Augmented Hooke's law in frequency Domain.
A three dimensional, material damping formulation.
Int. Journal Solids and Structures, Vol. 32 No. 19, pp. 2835-2852, 1995.

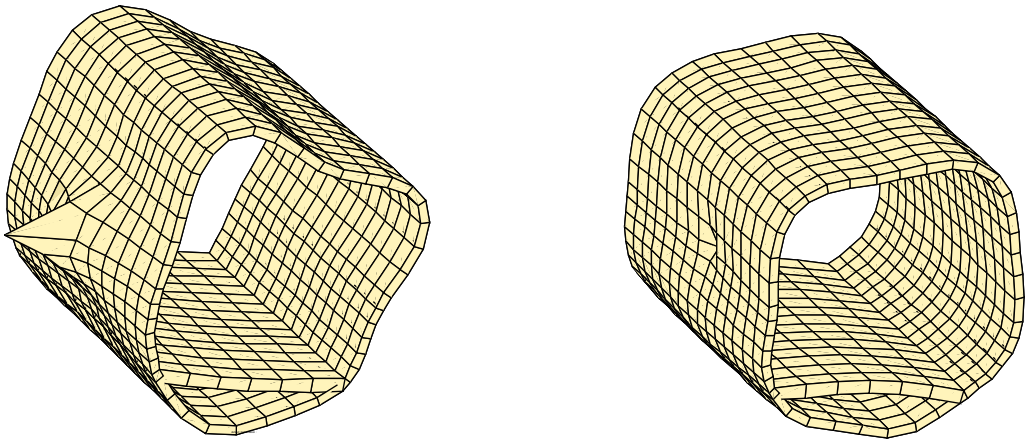
[7] **Lesieutre, G. A.**
Finite Element Modeling of Frequency Dependent Material Damping using Augmenting Thermodynamic Fields.
Ph. D. Thesis, University of California, Los Angeles, 1989.

[8] **Taylor R.L, Beresford P.J & Wilson E. L**
A non-conforming element for stress analysis.
Int. Journal for num. methods in engineering, Vol. 10, 1211-1219 (1976).

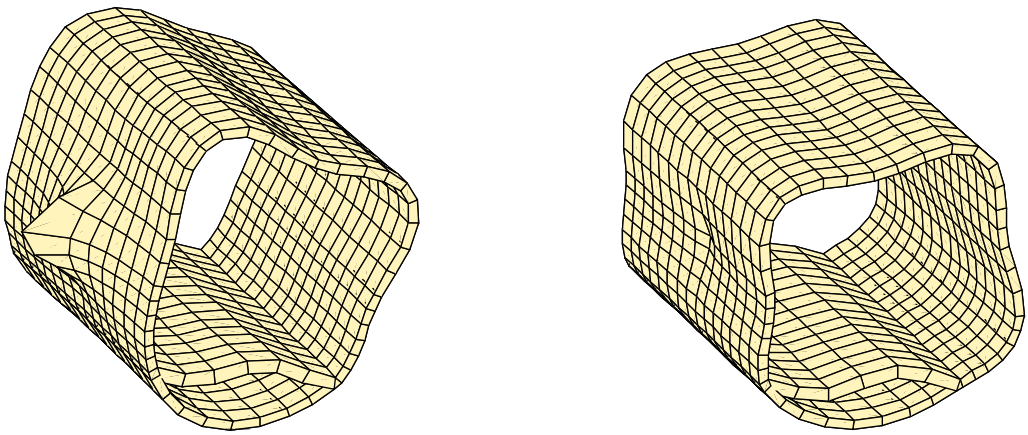
[9] **Gustavsson M, Hansson P.A**
Dynamic Analysis of a shell using 'Morley' and 'enhanced strain' elements.
Div. of structural Mechanics (internal document), LTH, Lund University, Lund, Sweden 1998.



Complete finite element model response for driving point '363R10'. Material parameters determined by the Inverse modeling using all, ***112,*** frequency response function columns.



Complete finite element model response for driving point '363R10'. Material parameters determined by the Inverse modeling using ***12*** frequency response function columns.



Complete finite element model response for driving point '363R10'. Material parameters determined by the Inverse modeling using ***6*** frequency response function columns.

Evaluation of Solution Methods for Finite Element Analyses of Vibrations and Acoustics

Mats Gustavsson

Div. of Structural Mechanics, LTH, Lund U niversity, Lund, Sweden

ABSTRACT

Finite element analysis of vibrations and acoustics involves finding the solution to linear equation systems. In most practical situations rather large finite element models are required to give sufficient quality of the analysis. Consequently, large equation systems arise. However, the system matrices generally are *sparse* i.e., only very few of the matrix entities are non-zero, allowing for significantly more compact storage and faster computations than would be the case for full matrices.

If damping is represented in the model, the equation system will be non-singular, but may have both positive and negative eigenvalues. Systems with both positive and negative eigenvalues are *indefinite*.

Several alternatives are available for solving large sparse systems of equations. Iterative solvers have some intrinsic attractive properties compared to direct solvers, especially in terms of memory usage and the possibility for efficient implementations in parallel computing environment.

This paper shows results from using the *Conjugate Gradient Method* for solving indefinite systems. In the literature (e.g., [1] and [2]) the conjugate gradient method is derived assuming positive definite matrices, but the result of this study shows that indefinite systems can also be considered for the conjugate gradient method.

Comparisons are made between direct solvers and the conjugate gradient method using two versions of pre-conditioning (Jacobi and Incomplete LDL^T-factorization). Implementation aspects, as well as computational environment considerations are discussed.

For the large acoustic systems used as an application example, the present implementation of the conjugate gradient method is generally faster than the direct solution methods used for comparisons.

For the structural dynamics examples, pre-conditioning is required to get convergence for the conjugate gradient method. This is so both for frequencies giving a positive definite system and frequencies leading to an indefinite system. Also, for these systems the possibility of using an iterative solution strategy is proven. The present implementation of the pre-conditioning (incomplete LDL^T) is far from optimal, resulting in long solution times. This implementation can certainly be improved, e.g. by using the concept of the direct LDL^T-solver, and then the iterative solver would not only require less memory but may also be faster.

1 Introduction

In this paper the *Conjugate Gradient method* is derived and compared to other common solution methods for large sparse systems of equations.

Iterative solution methods typically require significantly less memory but more computations than direct solution methods. However, the majority of the computations are matrix-vector multiplications, which are straight-forward to implement in a parallel computing environment. A proper implementation of iterative methods may therefore be an interesting alternative to direct solvers in a parallel computing environment.

The most popular iterative method is probably the conjugate gradient method. However, this method is derived assuming symmetric, positive definite systems, it is not recommended for indefinite systems. Certainly other iterative methods exist, as well as methods to re-formulate an indefinite system into a positive definite system. But it is questionable if an equally efficient iterative method can be found, and re-formulating to a positive definite system by the methods suggested in [1] requires knowledge of the extreme eigenvalues of the system and will reduce the conditioning of the system.

For some types of analysis, for example in a perturbation analysis to numerically evaluate derivatives, iterative methods can benefit from having an initial, approximate, solution. Further, when using so called pre-conditioning, the same pre-conditioning matrix may be applied to systems with reasonably small differences to enhance convergence speed and stability. For example, both perturbation analyses and frequency sweep analyses can take advantage of having an initial solution and a pre-conditioning matrix for an adjacent frequency.

The present implementations of the conjugate gradient method, with incomplete LDL^T-factorization for pre-conditioning are probably quite far from optimal concerning computational speed and memory usage. This should be kept in mind when comparing the solution times for the iterative concept with the highly optimized direct LDL^T and wave-front solvers.

Solution methods based on modal sub-structuring are not considered in this paper. There are situations where such methods are preferable with respect to computation times. However, modal methods with truncation of the number of modes to be used for the response analyses introduce an error which can be difficult to estimate. Not only the frequency range of the analysis and the excitation characteristics influence the proper selection of modes to be included. Models including representation of damping are also less straight-forward to analyze with modal methods.

2 Finite element formulation of dynamic systems

2.1 Dynamic equilibrium equation

Consider a general dynamic system

$$Kx(t) + \sum_n C_n \frac{d^n}{dt^n} x(t) + M\ddot{x}(t) = f(t) \qquad \text{[Eq. 1]}$$

where K is a stiffness term, M is a mass term and the sum over n constitute the forces related to damping mechanisms for the system. Viscous damping is represented by n=1, and more advanced damping representations by other values (possibly including non-integer values).

2.2 Frequency domain formulation

The force equilibrium may be represented in the frequency domain rather than in the time domain by Fourier transformation

$$F(\omega) = \int_{-\infty}^{\infty} (f(t)e^{-i\omega t})dt \qquad \text{[Eq. 2]}$$

and if a linear system is assumed the response (x) will have energy at the same frequencies as the force, i.e.

$$X(\omega) = H(\omega)F(\omega) \quad . \qquad \text{[Eq. 3]}$$

H(ω) is the *frequency response function* between response and force at frequency ω.

The frequency domain version of Eq. 1 is

$$KX(\omega_k) + \sum_n C_n \frac{d^n}{dt^n} X(\omega_k) + M\ddot{X}(\omega_k) = F(\omega_k) \qquad \text{[Eq. 4]}$$

and evaluating the time derivatives, leads to

$$\left[K + \sum_n (i\omega_k)^n C_n - \omega_k^{\,2} M\right] X(\omega_k) = F(\omega_k)$$

$$\text{or} \qquad H(\omega_k)^{-1} X(\omega_k) = F(\omega_k) \quad . \qquad \text{[Eq. 5]}$$

The system given by Eq. 5 is generally large and sparse, and will have negative eigenvalues for frequencies above the first resonance of the system. The time domain response may be represented by the integral of the frequency domain response at each frequency:

$$x(t) = \int_{-\infty}^{\infty} (H(\omega)F(\omega)e^{iwt})d\omega \qquad \text{[Eq. 6]}$$

if required.

If no damping mechanisms are included, the system matrix will become singular at the resonance frequencies, but all physical systems have damping, and including some damping mechanism will guaranty a non-singular system.

For the study of different solutions methods, a complex-valued, linear, symmetric and non-singular system

$$Ax = b \ ,$$

with A being an N by N matrix, x the solution vector (N by 1) for the right hand side vector b (N by 1), is considered. For the direct LDL^T and iterative solvers the equivalent real-valued form is used, but the system is still referred to as $Ax = b$.

3 *Direct solution methods*

3.1 Gauss elimination and LU-factorization

Gauss elimination and the related LU-factorization are probably the most well-known methods for solving linear systems. The general idea behind the Gauss elimination is to convert the general system $Ax = b$ into a triangular system $Tx = f$, where T is an upper triangular matrix. By simple back substitution the solution vector x is easily found for the triangular system.

For the alternative LU-factorization method, the matrix A is factorized $A = LU$, where L is a lower triangular matrix and U is an upper triangular matrix. The solution vector x is then found by solving the two triangular systems $Ly = b$ and $Ux = y$.

3.2 LDL^T- and Cholesky-factorization

If the system matrix A is symmetric, a factorization $A = LDL^T$, where L is a lower triangular matrix and D is a diagonal matrix, can easily be shown to exist. If A is positive definite, the effect of the diagonal matrix D can be captured in the triangular matrix L. This factorization is called Cholesky factorization ($A = Z^TZ$) and requires fewer operations than the LDL^T factorization. However, the systems resulting from finite element formulation of structural dynamic and acoustic systems are indefinite and the LDL^T formulation consequently has to be used.

In the present study a parallel implementation of the LDL^T factorization and the subsequent solution of the system is used (Silicon Graphics 'PSLDLT'-routines).

3.3 The 'wavefront' solution technique

For banded systems, methods based on successive elimination of variables - the so called 'wavefront' methods - can be efficient. A description of the wavefront technique can be found in [3]. In the present study a multi-frontal wave-front solver, implemented in the general finite element program ABAQUS, is used.

4 *Iterative methods*

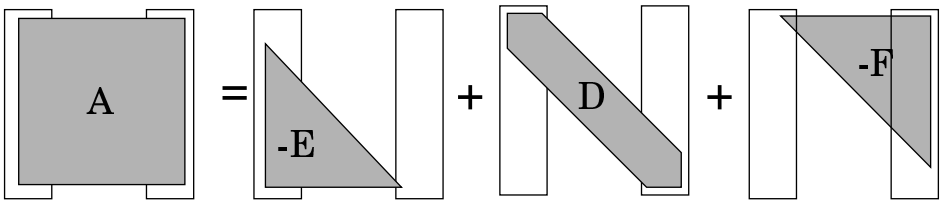
In some situations methods based on iteratively trying to find a solution to an equation system are attractive. Compared to the direct methods, e.g. the methods described in *Section 3*, iterative methods usually require less memory and are easier to implement in a parallel computer environment. On the other hand iterative methods do not always converge to t give the correct solution, and, even if convergence is achieved, the number of iterations required to get a solution with sufficient accuracy may be large. Furthermore, the performance of iterative methods is more sensitive to system characteristics than direct methods. Given these one of the main objectives of this study is to investigate how well iterative methods work for typical structural dynamics and acoustic systems.

See [1] and [2] for a more thorough description of iterative methods.

4.1 Jacobi, Gauss-Seidel and SOR/SSOR

The Jacobi method is a very simple iterative method that can be successful, in particular, for solving diagonally dominant systems.

Consider the system $Ax = b$. Split the matrix A into three parts; a strictly lower triangular matrix ($-E$), a diagonal matrix (D) and a strictly upper triangular matrix ($-F$).



A row, j, of the system $Ax = b$ can then be written

$$\sum_{i=1}^{j-1} -E_{ji}x_i + D_{jj}x_j + \sum_{i=j+1}^{N} -F_{ji}x_i = b_j \quad . \qquad \text{[Eq. 7]}$$

Pre-multiplying all terms with D_{jj}^{-1} we get

$$D_{jj}^{-1}\sum_{i=1}^{j-1} -E_{ji}x_i + x_j + D_{jj}^{-1}\sum_{i=j+1}^{N} -F_{ji}\,x_i = D_{jj}^{-1}b_j \qquad \text{[Eq. 8]}$$

or

$$x_j = D_{jj}^{-1}\sum_{i=1}^{j-1} E_{ji}x_i + D_{jj}^{-1}\sum_{i=j+1}^{N} F_{ji}x_i + D_{jj}^{-1}b_j \quad . \qquad \text{[Eq. 9]}$$

This explicit expression for x_j requires knowledge about the other terms in the solution vector x, which are obviously not available. However, Eq. 9 can be written in iterative form, giving a new estimate x_1^{p+1} based on a previous

estimate of the solution x_1^p . Writing Eq. 9 in this iterative form for all rows and returning to the matrix format, the Jacobi iteration algorithm is achieved:

$$x^{k+1} = D^{-1}(E+F)x^k + D^{-1}b \quad . \qquad \text{[Eq. 10]}$$

An intuitive improvement of the Jacobi algorithm can be obtained by returning to the component form Eq. 9, and writing it in iterative form

$$x_j^{p+1} = D_{jj}^{-1}\left(\sum_{i=1}^{j-1} E_{ji}x_i^p + \sum_{i=j+1}^{N} F_{ji}x_i^p\right) + D_{jj}^{-1}b_j \quad . \qquad \text{[Eq. 11]}$$

A natural choice is to let *j=1:N* and then a new estimate x_j^{p+1} is available and can be used instead of x_i^p when determining x_{j+1}^{p+1}. This concept, using the most recent x_j's as they become available, results in the Gauss-Seidel iteration scheme:

$$x^{k+1} = (D-E)^{-1}Fx^k + (D-E)^{-1}b \quad . \qquad \text{[Eq. 12]}$$

Contrary to the Jacobi method, the result of a Gauss-Seidel iteration step is dependent on the row ordering, i.e. if two rows are interchanged this will influence the results. The algorithm given by Eq. 12 is called the forward version of the Gauss-Seidel algorithm, making the updates in the order x_1-x_N. Reversing the order of the updates gives the backward version:

$$x^{k+1} = (D-F)^{-1}Ex^k + (D-F)^{-1}b \quad . \qquad \text{[Eq. 13]}$$

It is quite common to alternate forward and backward Gauss-Seidel iterations. This is referred to as symmetric Gauss-Seidel iteration.

A striking fact for the Jacobi and Gauss-Seidel iteration method is the absence of influence of x_j^p when determining x_j^{p+1}. Also it is easy to realize that the solution may vary drastically from one iteration to the next. A method called *relaxation* may be used to stabilize the solution by setting

$$x_j^{p+1} = \omega x_j^{p+1}(method) + (1-\omega)x_j^p \quad , \qquad \text{[Eq. 14]}$$

where $x_j^{p+1}(method)$ is the update given by the selected iteration method (e.g. Jacobi) and ω is a scalar value between 0 and 1. The matrix form of Eq. 14 based on forward Gauss-Seidel iteration is

$$x^{k+1} = (D-\omega E)^{-1}(\omega F + (1-\omega)D)x^k + (D-\omega E)^{-1}\omega b \quad , \qquad \text{[Eq. 15]}$$

which is the Successive Over Relaxation (SOR) method. A backward version is given by interchanging *E* and *F*, and for symmetric SOR (SSOR) one forward step is followed by one backward step to complete an iteration step.

4.2 Projection methods

Probably the most efficient iterative solution methods are based on *projections*.

A selected part of the theory for projection methods, and a brief derivation of the *Conjugate Gradient Method* is presented in this section.

First, again consider the system $Ax = b$. The N by 1 solution vector x for this system can be represented as a linear combination of vectors spanning the N-space. Let the column vectors of V span the N-space and let the row vector y contain the N scaling factors then

$$x = Vy \quad . \qquad \text{[Eq. 16]}$$

The scaling factors, y, can be determined by solving

$$AVy = b \quad . \qquad \text{[Eq. 17]}$$

However, this require about the same computational effort as directly solving $Ax = b$ and is consequently of little practical use. A more attractive approach is to try to find accurate solutions using a reduced base for the projection - i.e. to use a N by M projection matrix V with $M<N$. No exact solution is guaranteed for this case, but a sufficiently accurate solution may be found if a proper selection of projection vectors is used.

4.2.1 The Krylov subspace

Obviously, if an approximate solution to a system of order N is searched for in a specific subspace R^M, with $M<N$, the quality of the solution is dictated by the selection of the subspace R^M. The *Krylov subspace*, K^m, defined for a matrix A and a vector v by Eq. 18, plays a central role in defining a search subspace for iterative solution methods.

$$K^m = span\{v, Av, A^2v, ...,A^{m-1}v\} \qquad \text{[Eq. 18]}$$

An orthogonal set of column vectors may alternatively be used to define the Krylov subspace. This can be achieved by an *Arnoldi procedure*, e.g. according to the schematic algorithm give below.

Arnoldi procedure (using modified Gram-Schmidt orthogonalization)

[1] specify a starting vector v_1, $norm(v_1)=1$
[2] for $j=1,2,....,m$
[3] $w_j = Av_j$
[4] for $i=1,...,j$
[5] $h_{ij} = (w_j,v_i)$
[6] $w_j = w_j - h_{ij}*v_i$
[7] end
[8] $h_{j+1,j} = norm(w_j)$
[9] $v_{j+1} = w_j / h_{j+1,j}$
[10] end

where the notation (w_j,v_i) is used for the inner product of w_j and v_i ($w_j^T v_i$).

For a real valued symmetric matrix the Arnoldi procedure can be simplified

to a Lanczos procedure (e.g. see [2] pp. 174).

Algorithm 1: Lanczos procedure

[1] *specify a starting vector* v_1, $norm(v_1)=1$
[2] *Set* $\beta_1 = 0$ *and* $v_0 = \mathbf{0}$ *(vector)*
[3] *for* $j=1,2,\ldots,m$
[4] $w_j = Av_j - \beta_j v_{j-1}$
[5] $a_j = (w_j, v_i)$
[6] $w_j = w_j - a_j * v_j$
[7] $\beta_{j+1} = norm(w_j)$
[8] $v_{j+1} = w_j / \beta_{j+1}$
[9] *end*

This simplification is a consequence of the fact that most of the terms H_{ij} becomes zeros if the matrix is symmetric. In particular note the cancellation of the inner loop (over i) and the fact that only three vectors (v_{j-1}, v_j and w) have to be stored compared to the Arnoldi procedure where the entire N by M matrix H (the h_{ij} terms) has to be stored.

4.2.2 The Conjugate Gradient Method

One of the most used iterative method for solving linear systems is the *Conjugate Gradient Method*. The concept of this method is to iteratively improve the solution by projections on an increasing Krylov subspace, defined by Lanczos vectors, and to require the subsequent residuals to be orthogonal to the previous residuals, i.e.

$$\tilde{x}_m \in K^m$$
$$r_m = (A\tilde{x}_m - b) \perp K^{m-1} \qquad \text{[Eq. 19]}$$

Assume an initial solution x_0, where x_0 is an arbitrary N by 1 vector. The residual r_0 for this solution is

$$r_0 = Ax_0 - b \quad . \qquad \text{[Eq. 20]}$$

The initial solution may not be the optimal mapping of the correct solution on the vector x_0, and a first improvement of the solution may be to write

$$r_1 = A\alpha_0 x_0 - b \qquad \text{[Eq. 21]}$$

with α being a scalar. The optimal value of α is found by minimizing the 2-norm of the residual, equivalent to minimizing the square of the 2-norm given by

$$r_1^2 = (A\alpha_0 x_0 - b)^T (A\alpha_0 x_0 - b) \quad , \qquad \text{[Eq. 22]}$$

leading to

$$\alpha_0 = \frac{(Ax_0, b)}{(Ax_0, Ax_0)} \quad , \qquad \text{[Eq. 23]}$$

where $Ax_o = r_0$ may be used.

For the succeeding iterations the approximate solution is written

$$x_{j+1} = x_j + \alpha_j p_j \ , \qquad \text{[Eq. 24]}$$

with p_j being the new search direction given by

$$p_j = r_j + \beta_{j-1} p_{j-1} \ . \qquad \text{[Eq. 25]}$$

When $j=1$ no previous search direction exists and $p_1=r_1$. For $j>1$ the new search direction should be orthogonal to all the previous residuals. However, this can be achieved by only making the new search direction orthogonal to the most recent residual r_{j-1}, i.e.

$$(r_j + \beta_{j-1} p_{j-1}, r_{j-1}) = 0 \ , \qquad \text{[Eq. 26]}$$

yielding

$$\beta_{j-1} = \frac{-(r_j, r_{j-1})}{(p_{j-1}, r_{j-1})} \ . \qquad \text{[Eq. 27]}$$

The residual, r_{j+1}, for the new approximation x_{j+1}, is given by

$$r_{j+1} = A(x_j + \alpha_j p_j) - b = r_j - \alpha_j A p_j \ , \qquad \text{[Eq. 28]}$$

which should be orthogonal to the previous residuals. Again this is achieved by simply making r_{j+1} orthogonal to the previous residual r_j, giving

$$\alpha = \frac{(r_j, r_j)}{(A p_j, r_j)} \ . \qquad \text{[Eq. 29]}$$

Further, from Eq. 25 we have

$$r_j = p_j - \beta p_{j-1} \qquad \text{[Eq. 30]}$$

and the nominator of Eq. 29 can then be written

$$\boxed{\alpha = \frac{(r_j, r_j)}{(A p_j, p_j)} \ ,} \qquad \text{[Eq. 31]}$$

using the fact that Ap_j is orthogonal to p_{j-1}.

Similar identification can be used to simplify the expression for β, leading to

$$\boxed{\beta_{j-1} = \frac{(r_j, r_j)}{(r_{j-1}, r_{j-1})} \ .} \qquad \text{[Eq. 32]}$$

The core of the derivation made above is the simplification introduced when orthogonalization only has to be made to the previous residual vector. This is equivalent to the orthogonalization in the Lanczos algorithm and is a consequence of the symmetry of the matrix A. The arguments for this, and a more stringent derivation of the conjugate gradient methods is found in [2].

Bringing it all together, a version of the Conjugate Gradient Methods is achieved.

Algorithm 2: Conjugate Gradient Algorithm

[1] *determine an initial solution vector* x_0
[2] *Set* $r_0 = b - Ax_0$, $p_0 = r_0$, $j = 0$
[3] *while* $(r_j, r_j) >$ *tolerance*
[4] $\alpha_j = (r_j, r_j) / (Ap_j, p_j)$
[5] $x_{j+1} = x_j - \alpha_j p_j$
[6] $r_{j+1} = r_j - \alpha_j A p_j$
[7] $\beta_{j+1} = (r_{j+1}, r_{j+1}) / (r_j, r_j)$
[8] $p_{j+1} = r_{j+1} + \beta_{j+1} p_j$
[9] $j = j + 1$
[10] *end*

A possible problem for indefinite systems is that the denominator (Ap_j, p_j) may become zero when determining α. A possible work-around if this happens is to restart with a different initial solution vector. However, in practical situations it is probably very seldom that (Ap_j, p_j) becomes identical to zero, or close enough to cause numerical problems. In the present study several hundred analyses have been performed, using three different models and a large number of starting vectors, without having any problems in this respect.

5 *Pre-conditioning*

Iterative methods have the disadvantage of being more sensitive to the matrix conditioning than direct solution methods. However, the use of *pre-conditioning* can improve the convergence rate and stability significantly. The general idea is to find a suitable transformation of the system $Ax = b$, e.g. $M^{-1}Ax = M^{-1}b$, with $M^{-1}A$ being significantly better conditioned than A. The transformation matrix, M^{-1}, may range from the identity matrix to the inverse of A. In the first case no improvement of the convergence rate will result and in the latter the exact solution will be found at the first iteration.

A variety of methods can be considered to find a proper transformation, or pre-conditioning, of a matrix. The simplest ones generally only improve the convergence marginally, but are simple to implement and associated with a low computational cost. The more advanced pre-conditioning methods are more computationally expensive and typically require a larger storage space, but may result in a substantial improvement of the convergence speed and increased stability. In the present study, two methods were used; Jacobi pre-conditioning and incomplete LDL^T factorization. This selection represents two extremes with respect to computational cost and system conditioning improvement.

A more complete coverage of pre-conditioning methods can be found in [1] and [2].

5.1 Jacobi pre-conditioning

Pre-conditioning based on the basic iterative methods is quite simple in theory and implementation, but generally significant improvement of the convergence speed and stability is not achieved [2].
However, Jacobi pre-conditioning, simply a scaling to get unity on the values on the diagonal, may give some improvement and the associated computational cost is extremely low. In some situations a scaling by the row entity with the largest absolute value may be preferable, instead of scaling with the value on the diagonal. The computational cost of such pre-conditioning is similar to that of Jacobi pre-conditioning.

5.2 Incomplete LDLT factorization

The ultimate pre-conditioning is to use the inverse of the system matrix A, i.e. $M = A$, and the system to solve becomes trivial ($A^{-1}Ax = A^{-1}b$). From the theory of direct solvers it is known that the inverse of a non-singular matrix can be formed from a factorization. For a symmetric, non-singular matrix a factorization of the form

$$A = LDL^T \quad , \qquad \text{[Eq. 33]}$$

with L being a lower triangular matrix and D being a diagonal matrix, exists. The inverse of A can then be written

$$A^{-1} = L^{-T}D^{-1}L^{-1} \quad , \qquad \text{[Eq. 34]}$$

and with the triangular and diagonal structure of the matrices L and D the inversion of L and D can be performed at rather low computational cost. The factorization itself, however, is normally computationally expensive and often results in large storage requirements for the factorization matrix L.

Looking at the structure of the LDL^T factorization, given by the algorithm below, it is found that the rows of the lower triangular matrix L are gradually formed by the entities of A for the present row, and the entities of the previous rows in the L matrix.

Algorithm 3: LDL^T-Factorization

```
[1]   for col=1:N
[2]      for row=1:N
[3]         v(i) = L(col,row)d(row);
[4]      end
[5]      d(col) = A(col,col) - L(col, 1:col-1)v(1:col-1)
[6]      for row=col+1:N
[7]         L(row,col) = (A(row,col) - L(row,1:col-1)v(1:col-1)) / d(col)
[8]   end
```

In order to reduce the number of non-zero entities in the matrix L a dropping rule, ignoring terms in the matrix L, may be applied. This may result in a significantly more sparse structure of L, and the factorization may consequently become faster. This is called *incomplete factorization* and can be used with success as pre-conditioning for iterative methods.

Two classes of dropping rules are common:

- *dropping based on limited fill-in*
- *dropping based on by threshold value*

For the first class of dropping rule a pre-defined structure of the factorization matrix is used. Typically the structure of the system matrix is used to determine the structure of the factorization matrix. A common method is to use the same structure for the factorization matrix as the lower triangular

part of the system matrix. The knowledge of the structure of the factor matrix allows for efficient memory allocation and simplifies the factorization.

The second dropping rule is to drop entities of the factorization matrix based on their value. With this strategy a better control of which terms are to be dropped is employed.

Generally both dropping rules give substantial improvement of the convergence speed and stability and the selection of which to use depends on the particular problem in question. However, for this evaluation the method of dropping by value was employed, mostly because a complete factorization is achieved if a low threshold value is used. Using the complete LDL^T factorization as pre-conditioning will, apart from round-off errors, give an identity matrix as system matrix, and this provides a possibility to check the algorithms.

Modifying row 7 of Algorithm 3 to:

Modification for Incomplete LDL^T-Factorization

L_tmp = (A(row,col) - L(row,1:col-1)v(1:col-1)) / d(col)
if (abs(L_tmp) > threshold)
L(row,col) = L_tmp

gives an algorithm for incomplete LDL^T-factorization using a threshold dropping rule.

5.3 Pre-conditioned Conjugate Gradient

Including pre-conditioning in the conjugate gradient algorithm, the following algorithm is achieved

Algorithm 4: Pre-conditioned Conjugate Gradient Algorithm

[1] *determine an initial solution vector* x_0
[2] *Set* $r_0 = b - Ax_0$, $z_0 = M^{-1}r_0$, $p_0 = z_0$, $j = 0$
[3] *while* $(r_j , r_j) > tolerance$
[4] $\alpha_j = (r_j , z_j) / (Ap_j , p_j)$
[5] $x_{j+1} = x_j - \alpha_j p_j$
[6] $r_{j+1} = r_j - \alpha_j Ap_j$
[7] $z_{j+1} = M^{-1}r_{j+1}$
[8] $\beta_{j+1} = (r_{j+1} , z_{j+1}) / (r_j , z_j)$
[9] $p_{j+1} = z_{j+1} + \beta_{j+1} p_j$
[10] $j = j + 1$
[11] *end*

The inversion of M should be interpreted symbolically and is replaced, as usual, by a back- and forward-substitution in the implementation of the algorithm.

A full derivation of the algorithm given above, as well as alternative formulations, can be found in [2].

6 *Computational environment*

Evaluation of the feasibility of different solution strategies has to be performed in a relevant computational environment. Currently most finite element analyses of structural dynamics and acoustic systems are performed using commercial software running on UNIX workstations in the medium to higher range, or larger computer systems. The present study of solution methods for structural dynamics and acoustic analyses is aimed at this kind of computer platform.

6.1 Selection of solution methods to be compared

Basically two solution alternatives are evaluated and compared to the direct wavefront solver used by the wide-spread commercial finite element code ABAQUS. The two are:

- *A direct solver based on LDL^T-factorization*
- *The iterative Conjugate Gradient Method (using Jacobi and incomplete LDL^T factorization)*

6.2 Computer system

Two computer systems were used for the evaluation:

- *Silicon Graphics Octane with two R10 000/195 MHz processors and 256 MByte RAM (1 Gbyte RAM for some analysis).*
- *Silicon Graphics Origin 2000 with 46 R10 000/250 MHz processors and 26 GByte RAM*

On the Octane system one processor was used, while in the Origin system 1 to 24 processor(s) were used for the evaluation.

6.3 Implementations

6.3.1 The direct LDL^T solver

The PSLDLT library, kindly made available by Silicon Graphics International (SGI), is a very efficient package for solving sparse linear systems using LDL^T factorization. The factorization and subsequent solution steps are implemented to take advantage of SGI parallel computing environments.

A general rule for getting high performance when factorizing sparse matrices is to first reorder the matrix to minimize so called fill-in's during the factorization process. The "multi-level nested dissection ordering" reordering routine available in PSLDLT was used for all analyses with the PSLDLT routines. After this reordering, the matrices were more optimal than the original structure achieved by the PATRAN routine for bandwidth minimization.

Factorization is normally the most computationally expensive step, both in terms of memory usage and computation time. When the factorization matrices are available, the final step, consisting of solving two triangular systems and a diagonal scaling, is a performed at a low computational cost.

6.3.2 The Conjugate Gradient Method

A software library was also supplied by SGI for iterative solvers. However, it was found that this package did not allow negative values on the diagonal of the matrix and could not take advantage of an initial guess or available pre-conditioning matrices. Other alternative iterative solution packages are available for sparse systems, but since the conjugate gradient method is normally not considered for indefinite systems it was suspected that most other implementations were similar to the one made by SGI and would not work for indefinite systems. Consequently a version of the pre-conditioned conjugate gradient method was implemented. The 'sparse-matrix/general vector' multiplication function used in the SGI iterative solver was kindly made available by SGI and used in the present implementation of the conjugate gradient method. This resulted in essentially the same performance of the present implementation as the SGI conjugate gradient solver (when using Jacobi pre-conditioning).

6.3.3 Pre-conditioners

For this study Jacobi and incomplete LDL^T factorization have been used. The Jacobi pre-conditioning is trivial and easy to implement for high efficiency. Incomplete LDL^T factorization with a threshold criteria for fill-in acceptance is substantially more complicated to implement if high efficiency is required. This becomes very clear, below, comparing the execution time for the present implementation of the *incomplete* LDL^T pre-conditioning with the significantly lower execution time for the *complete* factorization using the PSLDLT library. A complete factorization, as performed in PSLDLT, should require both more computations and larger memory than an incomplete factorization. Modifying the PSLDLT factorization routine for incomplete factorization should result in even shorter execution time than the present, complete factorization. This should, however, be made by SGI to get as high efficiency as possible. Hopefully this study may inspire such development.

6.3.4 The Wavefront method (ABAQUS)

A multi-frontal parallel wavefront solver is available in the ABAQUS finite element software package. This solver was used for comparisons with the LDL^T and Conjugate Gradient solvers. The ABAQUS developers have several years of experience with wavefront solvers, and the implementation and tuning of the algorithm is probably very well done. This makes the ABAQUS wavefront solver suitable for comparisons with other solvers.

ABAQUS version 5.7.5 was used for this evaluation.

7 *Example problems*

7.1 Acoustic analysis of aircraft cabin

A model of a part of the interior air cavity of a Saab 340 aircraft with 114150 linear 3D elements and 121023 nodes was used to study the properties of the direct LDL^T, the wavefront, and the iterative conjugate gradient solvers.

System damping was represented by a complex-valued compression modulus for the air, giving approximately the damping properties found from measurements [4]. Since the implementations of the direct LDL^T method and the conjugate gradient method are made for real-valued arithmetic, the complex-valued system was written on the equivalent real-valued form. This does not, however, influence the performance noticeable but it emphasizes the fact that a complex-valued problem of size $\boldsymbol{N}$ with bandwidth $\boldsymbol{B}$ is equivalent to a real-valued problem of size $\boldsymbol{2N}$ with a bandwidth of $\boldsymbol{2B}$.

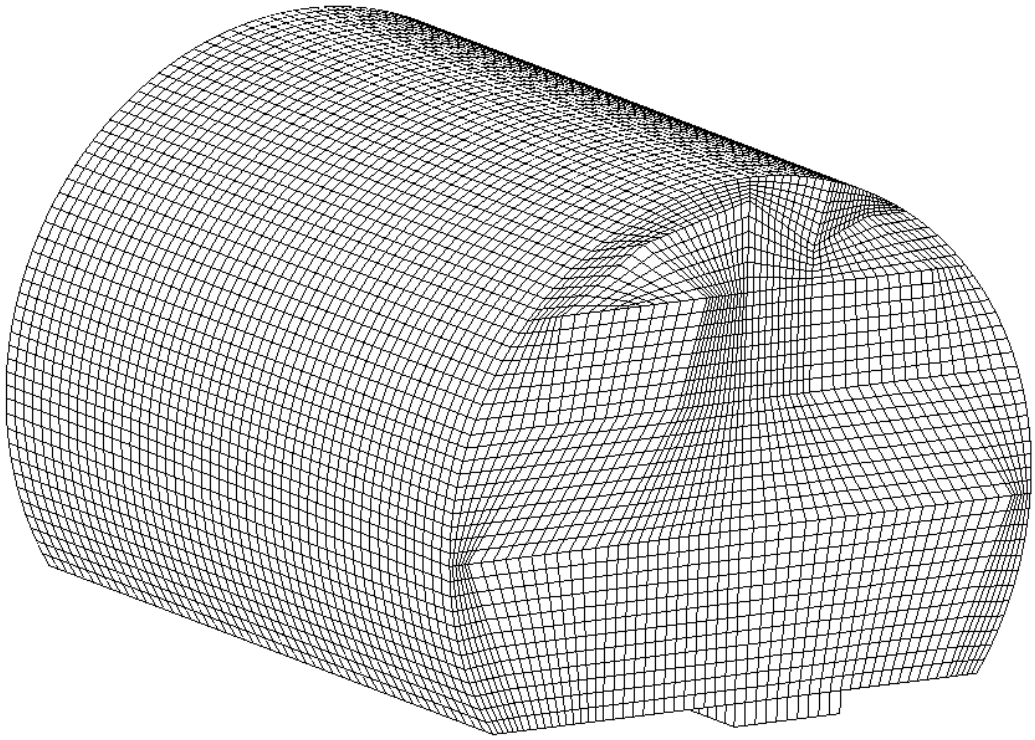


Figure 1: Finite element model of Saab 340 section (114150 elements, 121023 nodes).

The solutions are actually not of particular interest for this evaluation of solution methods, which is focused on the computer system requirements and solution times. However, it is important to have realistic models for the evaluation and consequently some examples of model responses are relevant. In *Figure 2* the acoustic responses at 82 Hz for a 1 dm^2 vibrating surface (indicated in the figure) is shown. In *Figure 3* three of the resonance mode shapes are shown together with a global frequency response curve.

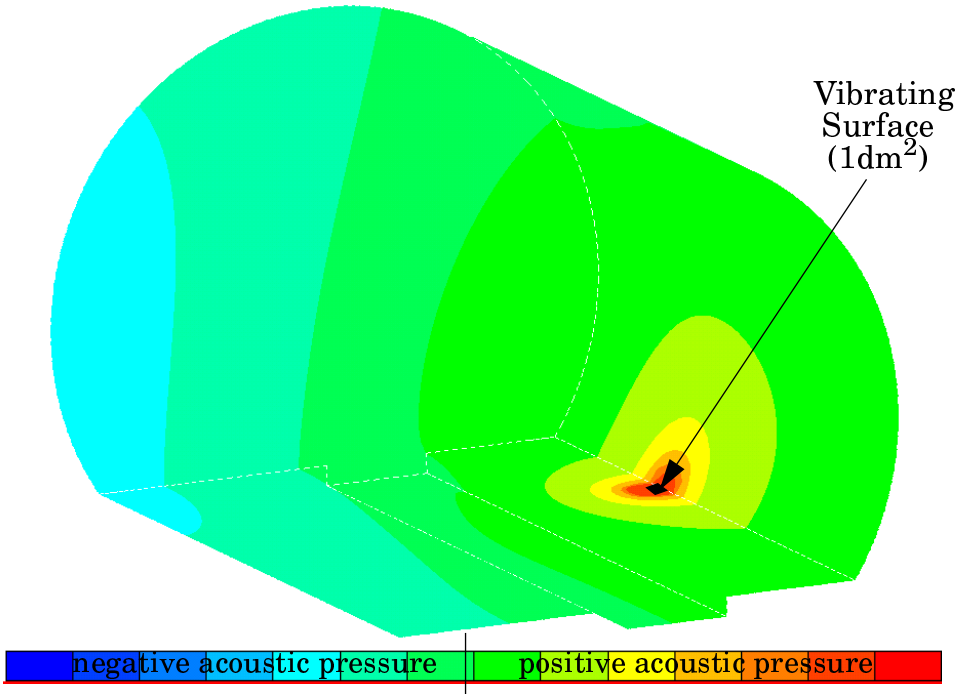


Figure 2: Response at 82 Hz (Blade Passage Frequency of the Saab 340).

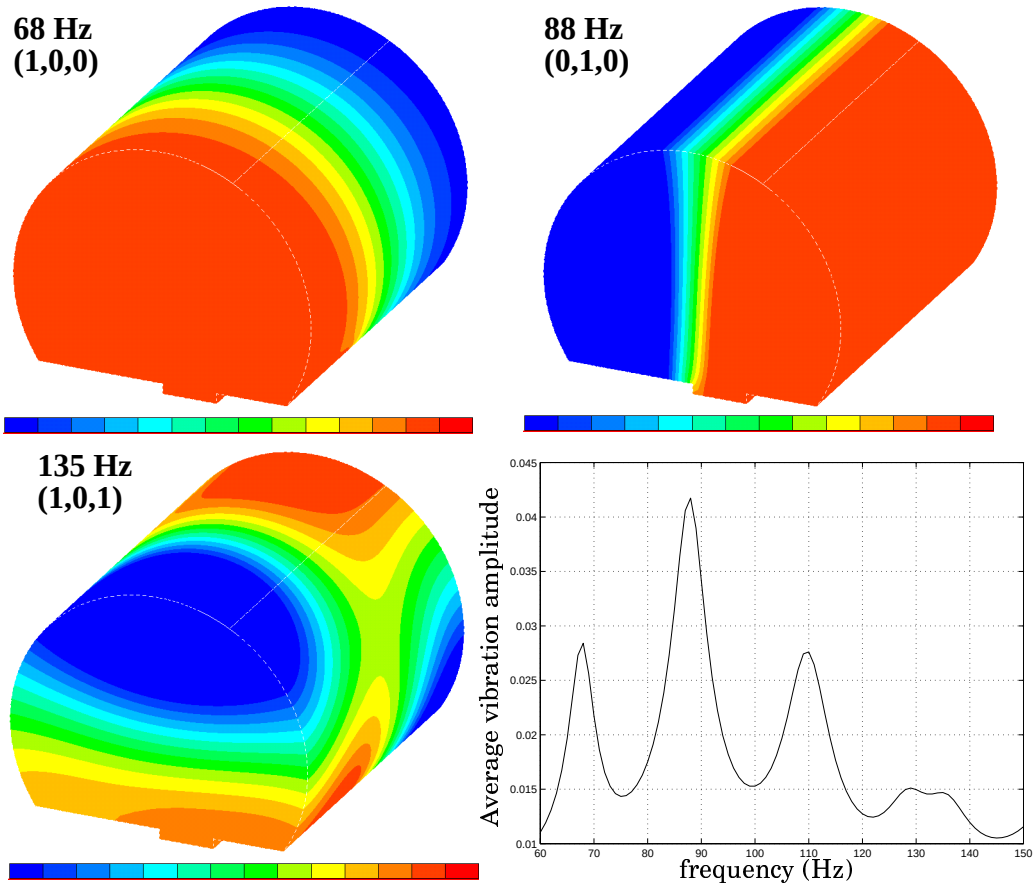


Figure 3: Acoustic resonances at 68, 88 and 135 Hz.

7.1.1 Performance of the direct LDL^T solver

This model was found to be too large to be solved practically with the direct LDL^T solver on the available Octane platform with 256 Mbyte of memory. During the solution process 2.5 GByte of memory is required, and that is more than what is usually available on a workstation. Swap space allows for using more memory than the RAM of the computer system. However this normally reduces the performance significantly and was not considered for the present problem.

Using one processor on the Origin system, 30 seconds was spent on reordering the matrix, 51 min on factorization, and about 30 seconds on the final solution of the triangular systems. One of the main characteristics of the PSLDLT solver is the performance increase with number of processors. This is clearly indicated by the results for 1 to 24 processors (*Figure 4*).

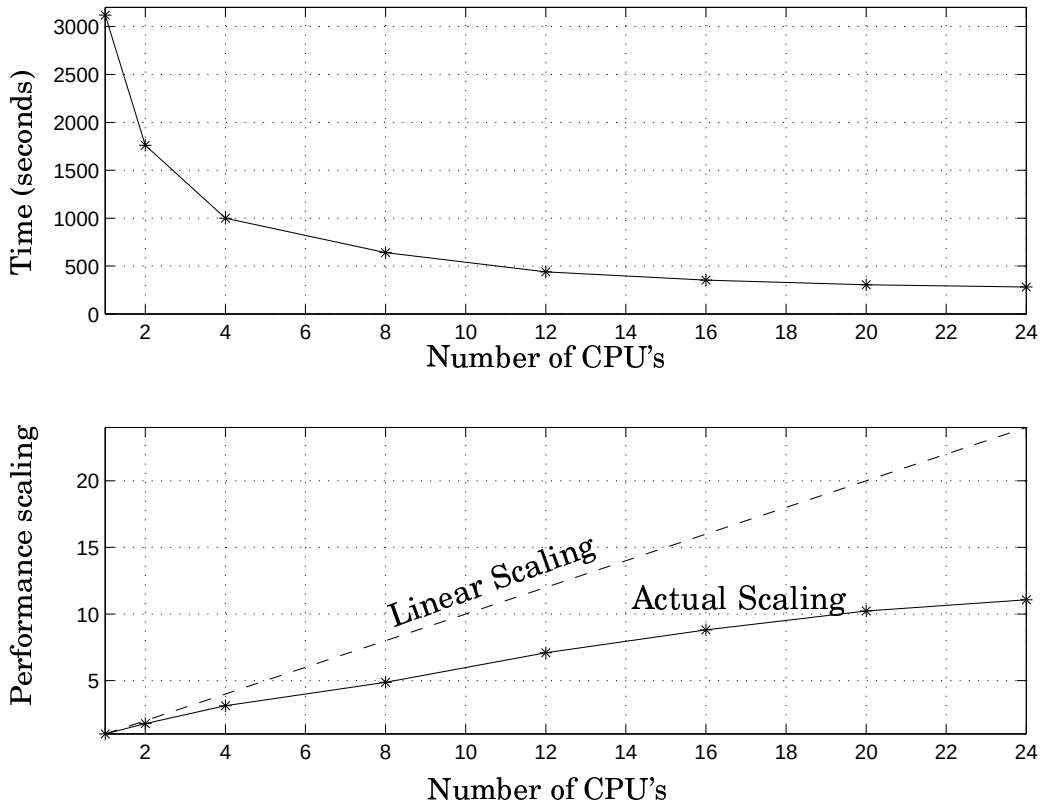


Figure 4: Solution times for PSLDLT using 1-24 processors (Origin 2000 R10 000 / 250 MHz).

For this kind of problem a linear scaling (indicated in the figure above) is practically an upper bound for the achievable speed-up. For more than 4 processors the achieved speed-up is essentially half the number of processors used. This consistency in speed-up is surprisingly good, particularly for the higher number of processors.

7.1.2 Performance of the conjugate gradient solver using Jacobi pre-conditioning.

Using the present implementation of the conjugate gradient method, with Jacobi pre-conditioning, the acoustic problem can be solved using the Octane workstation. However, since no comparison could be made with the direct LDL^T solver for the Octane platform, only the Origin was used also for the iterative solver. The total memory required is 350 MByte, which is substantially less than the 2.5 GByte used by the direct LDL^T solver. The convergence criterion was specified as tolerating a residual of 1e-9 of the norm of the applied "force". It was found that this criterion gave a very similar solution to the direct solvers and the convergence was very rapid after getting down to a residual of about 1e-4.

The solution time for one CPU is significantly less than for the direct LDL^T solver, but the improvement when using more processors is small (*Figure 5*).

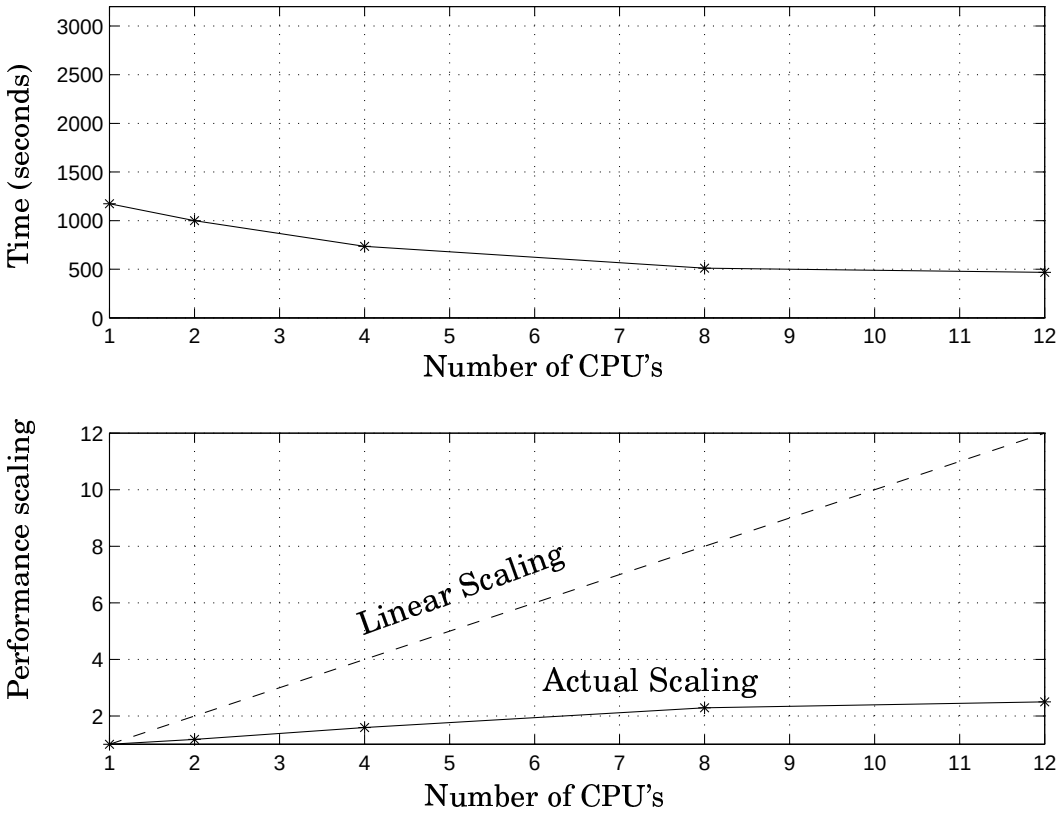


Figure 5: Solution times for the conjugate gradient solver using 1-12 processors (Origin 2000 R10 000 / 250 MHz).

This is not typical for iterative solvers which in principle can be implemented to scale almost linearly, since the majority of the computations are matrix vector multiplications that can be executed in parallel. Further tuning of the implementation, in particular concerning memory aspects, should allow for significant improvement for multi-processor execution. However, using fewer than 12 processors the iterative solver is faster than the direct LDL^T-solver.

7.1.3 Performance using the wavefront solver

The performance of the ABAQUS wavefront solver was consistently only evaluated for the Origin platform. With recommended memory settings for the wavefront solver, 500 MByte of RAM and 2.75 GByte disk space is used during the solution process. The execution times for the wavefront solver, using 1-12 CPU's on the Origin system, are given in*Figure 6*

.

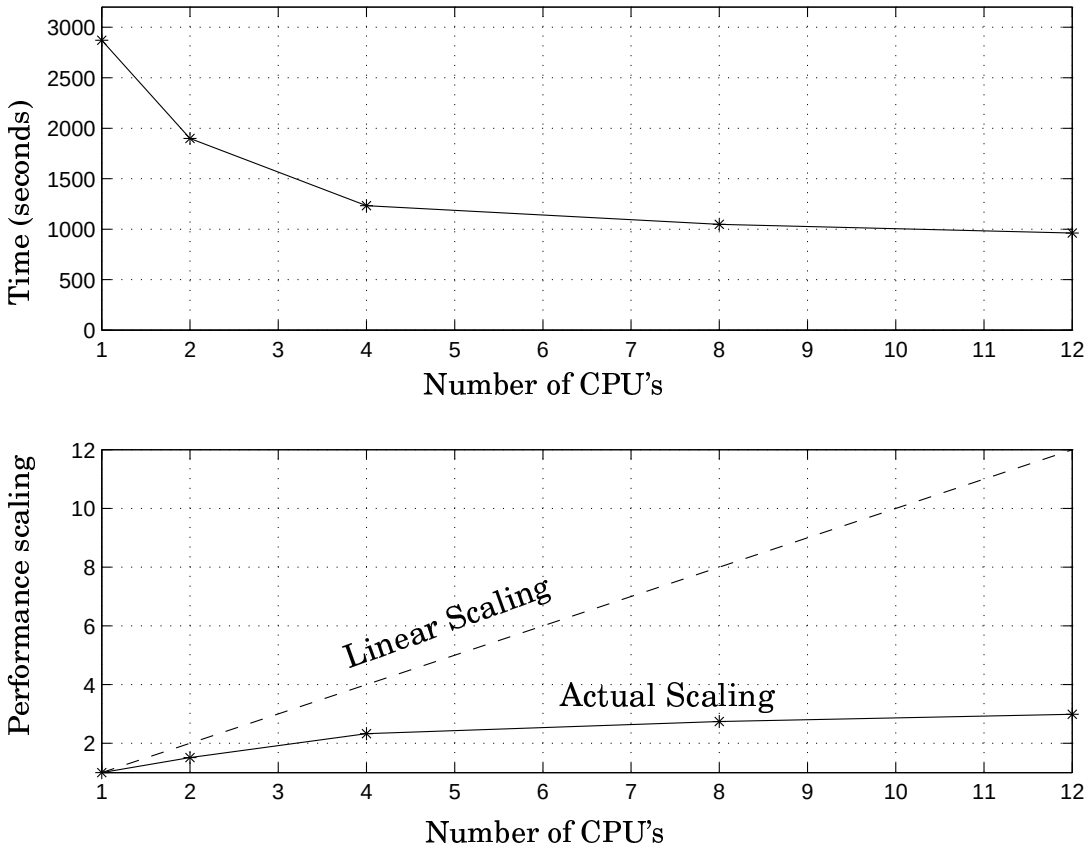


Figure 6: Solution times for wavefront solver using 1-12 processors (Origin 2000 R10 000 / 250 MHz).

When using one processor the wavefront solver is slightly faster than the direct LDL^T-solver, but the improvement with number of processors is significantly less, making the LDL^T-solver about twice as fast for 12 processors. The wavefront solver has a speed-up for multi-processor execution similar to the conjugate gradient solver, although the iterative solver is significantly faster at any number of processors.

7.1.4 Frequency sweep analyses

Frequency sweep analyses are common in structural dynamics and acoustics for identifying resonances and the variation in response over a frequency range. The frequency spacing is selected to give smooth curves, i.e., small variations in response for two adjacent frequency lines. For iterative methods this can be used to supply an initial solution. *Figure 7* shows a comparison of solution times for the two cases, with and without supplying an initial solution, for a frequency sweep analysis. An interesting fact is the increased solution time with frequency, which is related to the change in conditioning of the system matrix.

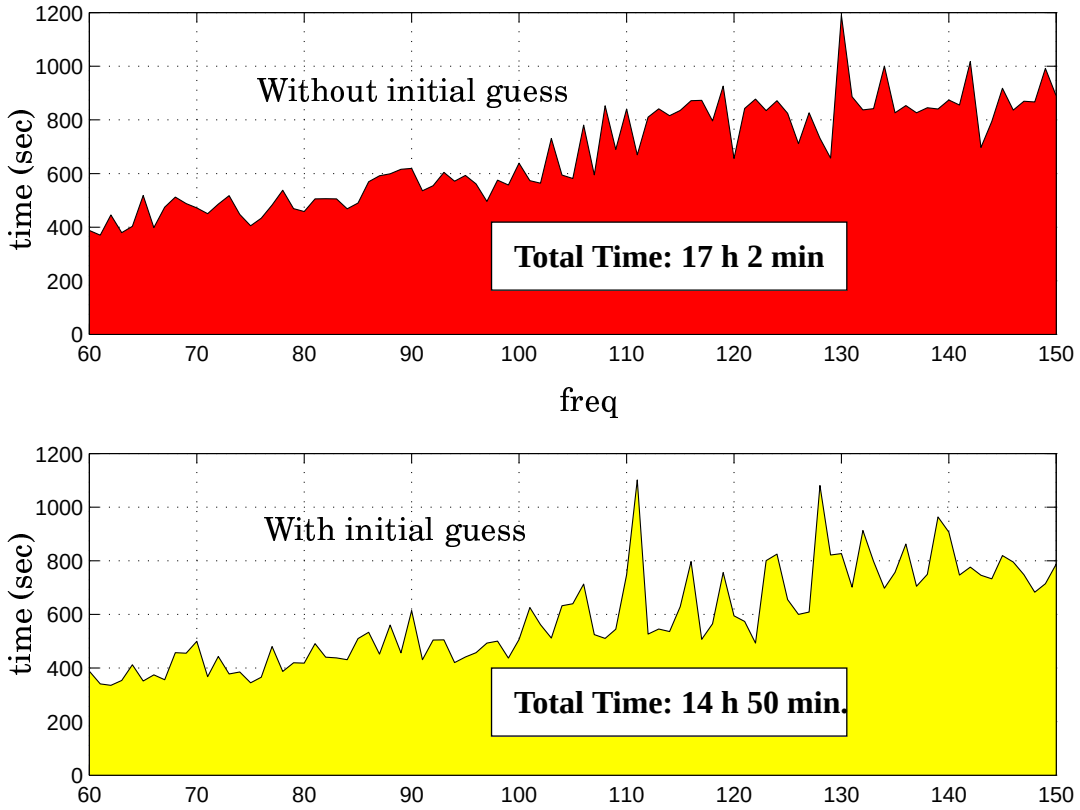


Figure 7: Solution times without initial guess (upper) and using the solution from previous frequency line as initial guess (lower) using 8 CPU's.

If a more computationally expensive pre-conditioning is applied it may also be possible to use the same pre-conditioning for analysis at several frequencies. This was investigated using incomplete LDL^T factorization at 82 Hz (threshold 1e-3) for the frequency range 60-100 Hz. The number of non-zeros in the L-matrix for this case is about 15 million and the complete factorization can be stored in 181 MByte. The present implementation of the incomplete LDL^T-factorization can be considered as a first version and not optimal in efficiency. The computation time for the factorization is therefore long (about 7 hours). Significant reduction of the solution times was achieved using incomplete LDL^T pre-conditioning, compared to using Jacobi pre-conditioning, not only at 82.0 Hz but in the 60-100 Hz frequency range (*Figure 8)*.

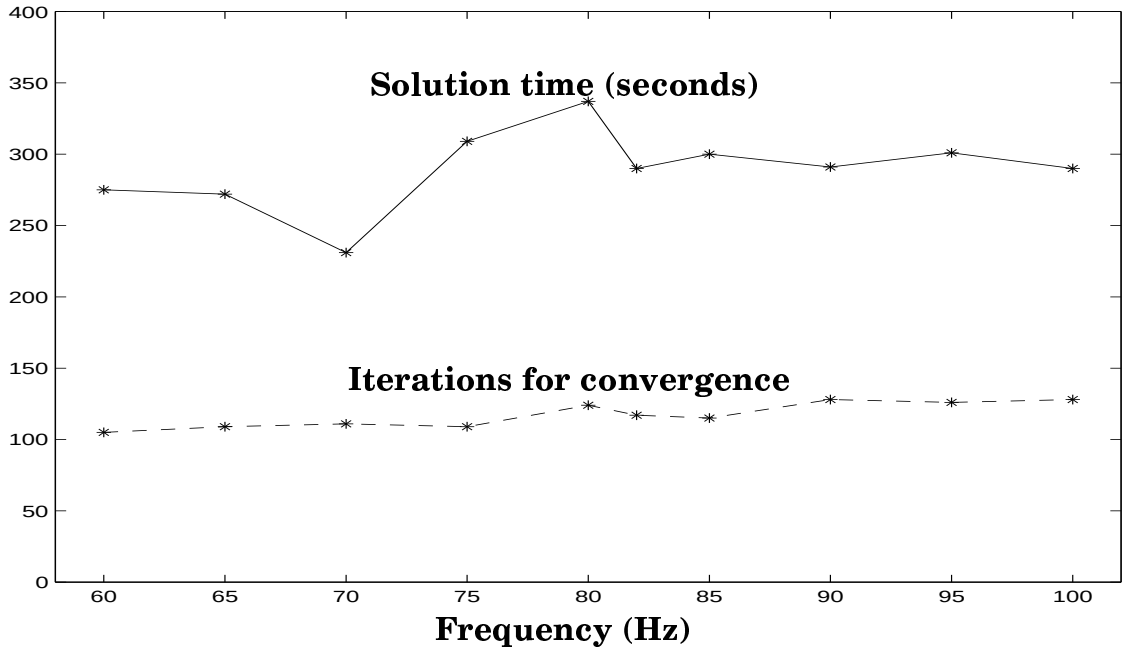


Figure 8: Solution times in seconds (solid), Number of iterations (dashed). (ILDL^T at 82Hz).

The variation in the solution time is more an indication of the load on the Origin system than an indicator of solution time variation with frequency, as can be seen on the number of iterations required for convergence. However, a slight increase in solution time (and required number of iterations) with increasing frequency is noticeable.

For a frequency sweep from 60 to 150 Hz (with Δf=1.0Hz) run in a one CPU environment, the table below [*TABLE 1*] shows the total execution times, the memory requirements and the disk requirements for the solution alternatives discussed in this paper.

TABLE 1. Frequency sweep solution time comparison for a one CPU Octane.

Method	Total time	Memory Req.	Disk-space
Conjugate Gradient (Jacobi pre-cond.)	33 h	350 MByte	-
Conjugate Gradient (Incomplete. LDL^T)	14 h (factorization: 7h)	700 MByte	181MByte (LDL^T-factors)
Direct LDL^T	78 h	2.5 GByte	-
Wavefront	72 h	500 MByte	2.75 GByte

It should, however, be noted that this is for a single excitation condition, which is favorable for the conjugate gradient method. On the other hand, the implementation of LDL^T and wavefront methods is more optimized and tuned than the implementation of the conjugate gradient method and the pre-conditioners.

7.2 Structural dynamics analysis (small model)

Two structural models of the Saab 340 aircraft were used for evaluation of solution methods. The first is a rather small model with 4512 dof's (*Figure 9)*. This model was originally developed and used for inverse modeling [11].

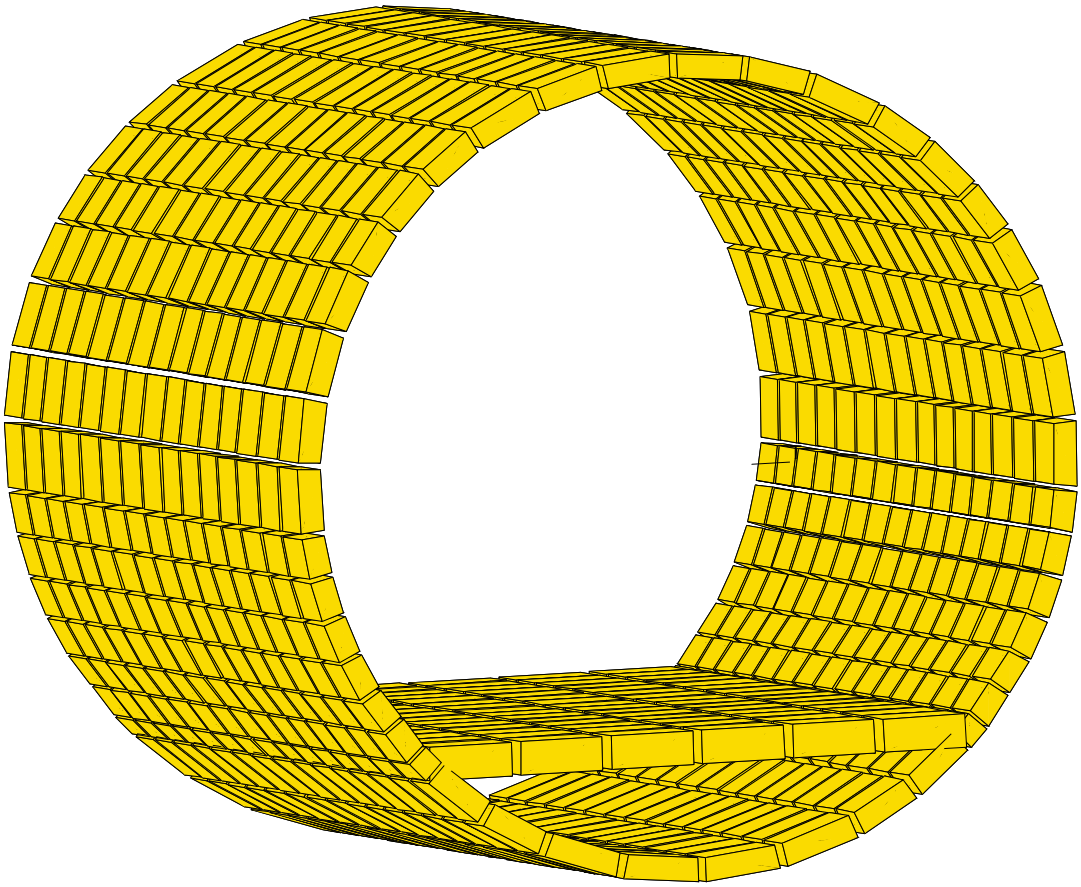


Figure 9: The small structural model with 720 elements and 1504 Nodes (4512 complex-valued dof's).

Again the variables are complex-valued, but still the problem is quite small (9024 real valued variables). There is a significant difference between acoustic analysis, like the one in *Section 7.1*, and analysis of vibrations for a model of shell structure. The structural system will have a poorly conditioned system matrix and hence will be more difficult to solve with iterative solvers.

To give an impression of the model behavior, the response for a 82.0 Hz point force is shown in *Figure 10*.

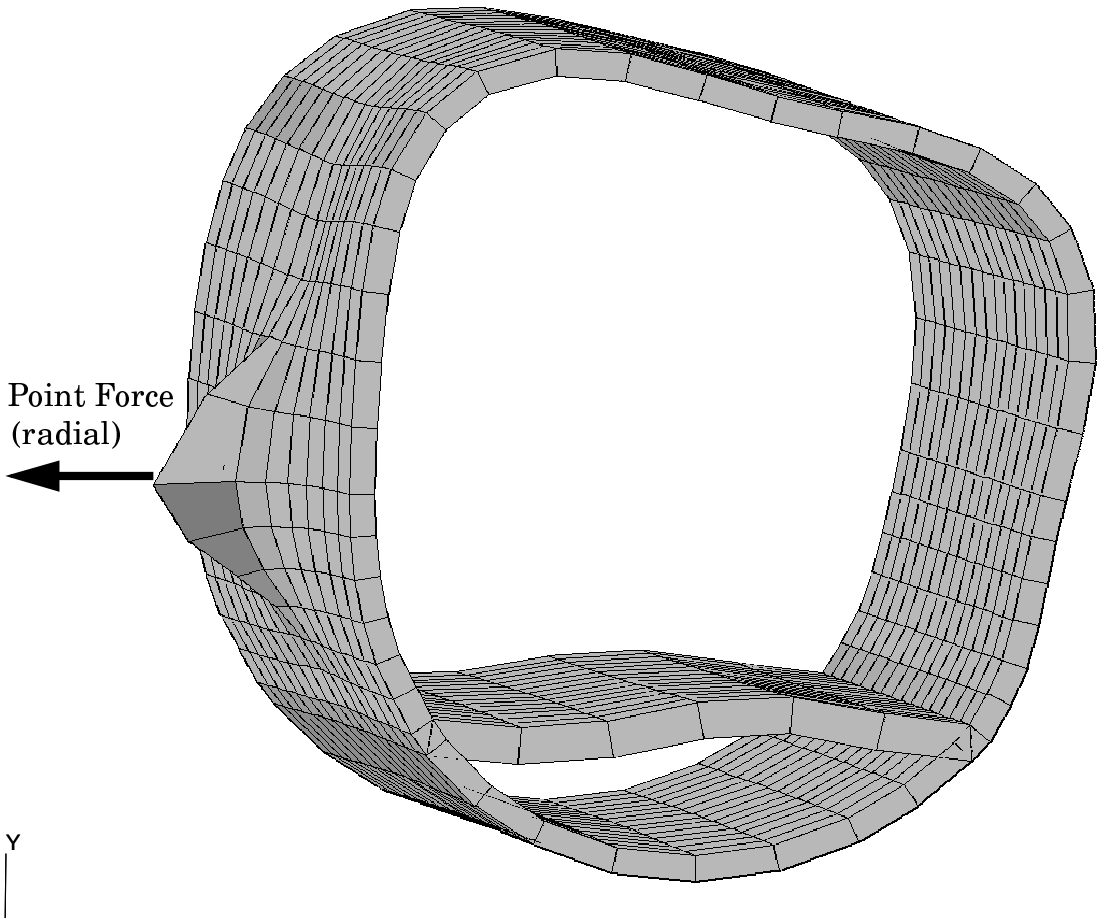


Figure 10: Response for a 82.0 Hz point force.

7.2.1 Performance of the direct LDL^T solver

The direct LDL^T-solver is very efficient for this problem with a total solution time of 4.5 seconds using one processor on the SGI Octane. The memory usage is also very low, and evaluation of the performance using multiple processor platforms is evidently not interesting.

7.2.2 Performance of the wavefront solver

The wavefront solver was not used for this example since it is likely to show very similar properties to the direct LDL^T solver for this small system.

7.2.3 Performance of the conjugate gradient solver

For the conjugate gradient method this problem is more of a challenge due to the poor conditioning of the system matrix. With Jacobi pre-conditioning the conjugate gradient method did not converge. This is caused by the rather poor conditioning of this system. Even for frequencies below the first resonance, when the system is positive definite, the Jacobi pre-conditioned conjugate gradient method did not converge.

For incomplete LDL^T-factorization, convergence was achieved for a threshold level of approximately 10-3 or lower. Typical performance for this case is given in the table below.

TABLE 2. Execution times using a one CPU Octane work-station

$ILDL^T$ threshold	**Time for $ILDL^T$**	**Iterations for convergence**	**Iteration time**
1e-3	63 sec	400	77 sec
1e-4	107 sec	20	6 sec
1e-5	144 sec	10	3 sec

For single right-hand sides, i.e. one excitation condition, the direct LDL^T solver is substantially more efficient than the pre-conditioned conjugate gradient method for this problem. One explanation for this is the more optimal structure achieved by the PSLDLT-routine for minimizing the number of non-zeros in the LDL^T-factorization compared to the original matrix structure which is not optimal in this respect. The original ordering of the matrix rows and columns was made in PATRAN using 'Optimize' with the option 'Cuthill-McKee / RMS Wavefront', while the PSLDLT pre-processor was set to use 'multi-level nested dissection ordering' [12]. The number of non-zeros for an LDL^T factorization of the original matrix is more than 4 million, while the PSLDLT-routine reports that just above 2 million non-zeros are required after the PSLDLT internal reordering. Certainly a more efficient implementation of the incomplete factorization itself can also be made.

For the acoustic test example it was found that a pre-conditioning matrix extracted at one frequency could be used in a rather broad frequency region around that frequency. The same was tried for the structural model and the results are shown in *Figure 11* (next page).

This example shows that the pre-conditioned conjugate gradient method can be used to solve structural dynamics problems. The actual model is too small to bother searching for alternatives to the direct solution methods. However, this small model was very useful for finding the required pre-conditioning and for the frequency sweep analysis using one incomplete LDL^T-factorization as pre-conditioning over the frequency band.

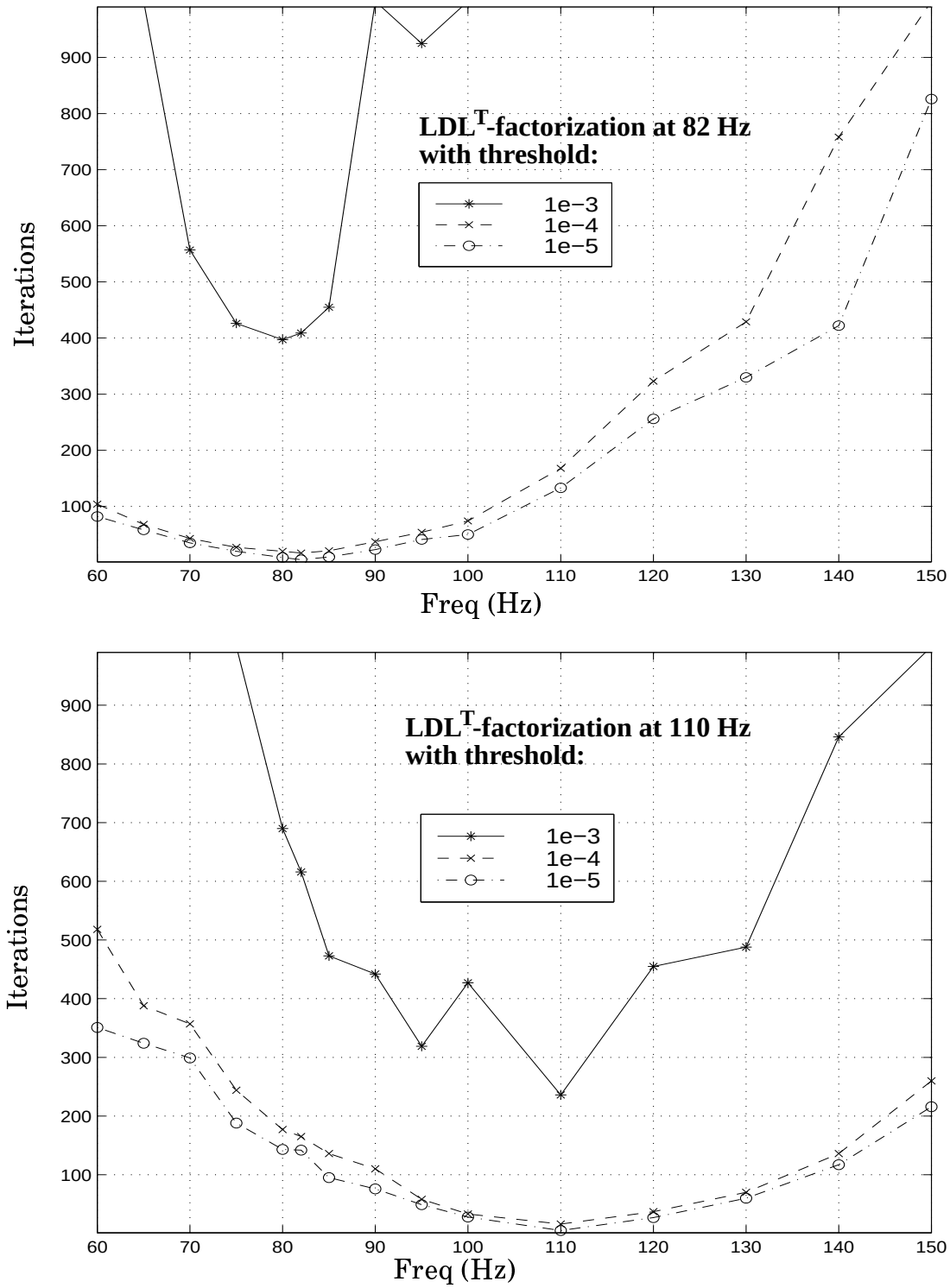


Figure 11: Number of iterations required for convergence using incomplete LDL^T-factorization with different thresholds at 82Hz (upper) and 110 Hz (lower) as pre-conditioning. If convergence is not achieved after 1000 iterations the solution diverges.

7.3 Structural dynamics analysis (large model)

A second structural model, more relevant in terms of number of elements for vibration analyses of an aircraft cabin structure (*Figure 12)*, was used to further evaluate the performance of the three solvers.

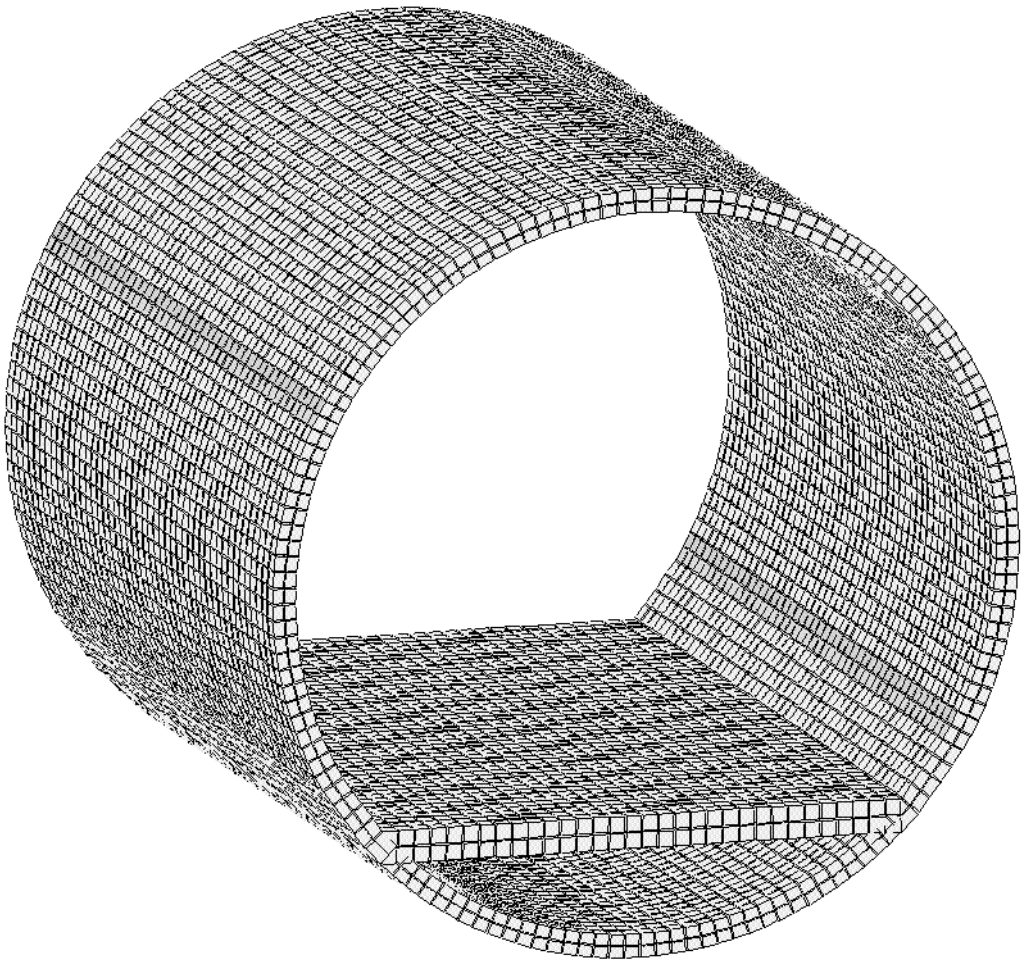


Figure 12: The large structural model with 16 544 elements, 25 200 Nodes (75 600 complex-valued dof's)

It can be argued that even larger models are required for analysis of complete aircraft structures and if analyses in the mid-frequency range are required. As can be seen by the results in the following section this model size is by no means the upper limit for the methods evaluated using the Origin 2000.

The model, with its 75 600 complex-valued variables, is considered to be large enough to be representative for practical situations, but not too large for efficient evaluation of solution methods.

7.3.1 Performance of the direct LDL^T solver

A total of 800 MByte of memory is required for solving this model with the PSLDLT routines. This is well below the maximum 2 Gbyte memory capacity of the Octane hardware, but more than presently available in the Octane used for this evaluation. Consequently, only the Origin platform was used for this model.

The total solution times for 1, 2, 4 and 8 processors are given by *Figure 13*.

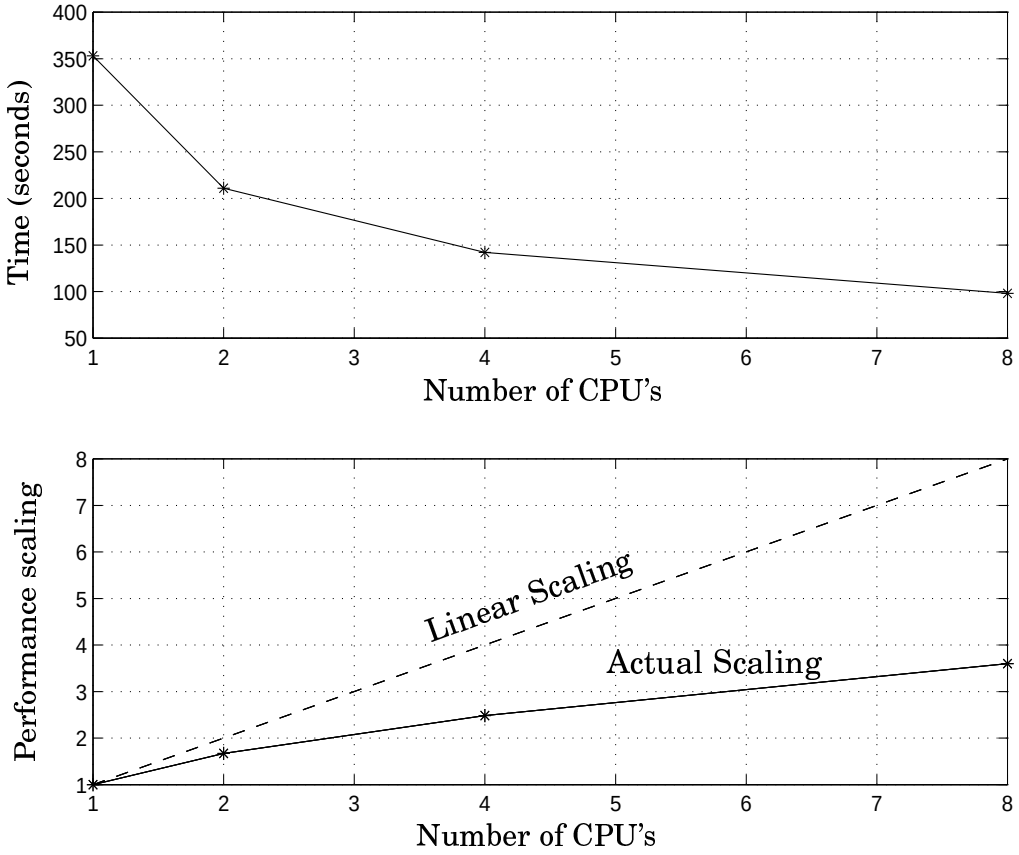


Figure 13: Solution times for PSLDLT using 1-8 processors Origin 2000 R10 000 / 250 MHz.

About 20 seconds was spent on pre-processing (matrix re-ordering), which is independent of the number of processors used. The majority of the remaining total solution time was spent on factorization, while the time for the back and forward substitution was negligible in comparison.

The performance scaling with number of processors is similarly impressive as for the acoustic model.

7.3.2 Performance of the conjugate gradient solver

As for the smaller structural model the conjugate gradient method did not converge when using Jacobi pre-conditioning, due to the poor conditioning of the system. Using incomplete LDL^T factorization as pre-conditioning, convergence was achieved when a dropping threshold of 1e-4 or lower was used.

The ordering of the matrix rows and columns made in PATRAN is, also for this model, far from optimal, compared with the ordering achieved with the PSLDLT pre-processor. For example, the number of non-zeros in the L-matrix for the incomplete LDL^T-factorization witha drop threshold of 1e-5 is more than 107 million, compared to about 86 million for the full factorization after the PSLDLT pre-processing. Not only the pre-conditioning factorization itself, but also the iterations made for the pre-conditioned system, is highly affected by the number of non-zeros in the factor matrix L. This is particularly evident when the systems are large and use large amounts of memory. For example the used ORIGIN 2000 system used here is equipped with 512 MByte on each CPU-card and if more memory is required additional 'overhead' time is spent when accessing memory on a different CPU-board. The present implementation of the preconditioned conjugate gradient method use about 3.5 GByte of memory for this problem. This means that at least 7 CPU-boards are involved in the process of supplying data for the iterative solver.

The above remarks should be kept in mind when examining [*TABLE 3*], giving some properties of the conjugate gradient solver for the large structural model.

TABLE 3. Approximate execution times and number of iterations required for convergence (out of balance force 1e-6 of applied force) for large structural model on the ORIGIN 2000 system using one CPU.

$ILDL^T$-threshold	Time for $ILDL^T$	Iterations for convergence	Iteration time
1e-4	~ 8 h	885	~ 200 min
1e-5	~ 10 h	34	~ 12 min

Compared to solving this problem with the PSLDLT functions, the iterative method is not a competitive alternative. In order to match the solution times for the PSLDLT solver, the pre-conditioned conjugate gradient method needs a more efficient incomplete LDL^T pre-conditioner implementation. Probalby the best way to get a more efficient incomplete factorization routine is to start with the complete LDL^T factorization in the PSLDLT library and make the necessary modifications. Certainly the incomplete factorization is a simplification of the complete factorization and should be possible to perform faster than complete factorization.

Similar to the analysis of the acoustic model, a frequency sweep analysis was performed using the incomplete LDL^T factorization at one frequency as pre-conditioning for the whole frequency band. The analysis frequency range was 60 to 150 Hz and an incomplete LDL^T-factorization at 82 Hz with threshold

1e-5 and 1e-6 was used. The number of iterations required for convergence at each frequency and for the two levels of factorization threshold is shown in *Figure 14*.

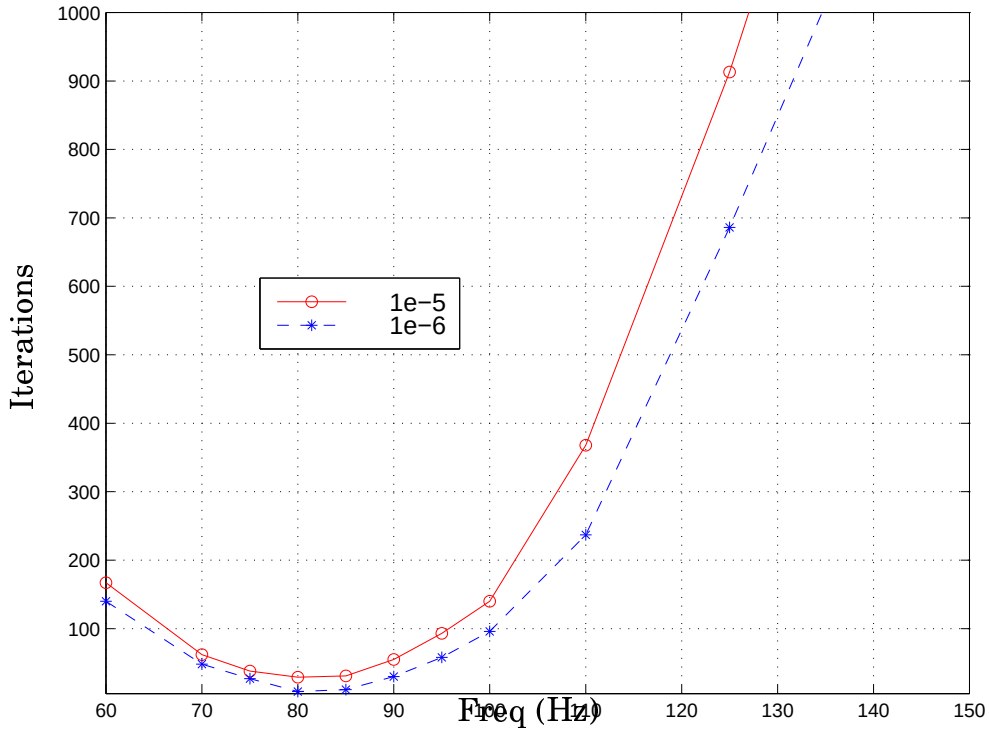


Figure 14: Frequency sweep using 82 Hz pre-conditioning.

At 150 Hz the solution did not converge for the maximum specified number of iterations (1000). For the present model it is clear that the lower threshold for the incomplete factorization results in faster convergence. The concept of using a factorization for a different frequency than the analysis frequency works as long as the two frequencies are reasonably close. However, more work has to be done on efficient implementation of the algorithms to compete with the PSLDLT direct solver in terms of solution speed and memory usage.

7.3.3 Performance of the wavefront solver

When using one and two processors the wavefront solver gives about the same solution times as the direct LDL^T solver. Similar to the results for the acoustic model, the wavefront solver did not benefit as much as the direct LDL^T solver using more processors. For example, the solution times for the wavefront solver using 4 and 8 processors were 247 and 234 seconds respectively, compared to 142 and 98 seconds for the direct LDL^T solver.

One advantage of the wavefront solver is the low memory requirement (about 185 MByte). On the other hand 1GByte of disk space is required for solving this problem with the wavefront solver.

8 Conclusions

The selection of a solution method for finite element analysis of vibrations and acoustics not only determine the analysis time, but also whether the problem is even possible to solve in a particular computer environment.

Available memory and disk space may restrict the user to a particular solution method or the use of a smaller model, in terms of variables, than desirable for accurate analysis results.

The present study shows the possibility of using the conjugate gradient iterative method for indefinite systems resulting from finite element formulation of structural dynamics and acoustic problems. Acoustic problems are solved faster with the present implementation of the conjugate gradient method than with the two direct solvers solution, unless more than 8 processors are available, when the direct PSLDLT solver is faster. For structural dynamics problems pre-conditioning is normally required for convergence. In this study an incomplete LDL^T-factorization implementation was made for pre-conditioning. It is shown that this pre-conditioning method is appropriate for the type of problems considered, but the implementation is not efficient enough to give solution times comparable with the direct solvers. There is no question that more efficient implementations of the incomplete LDL^T factorization can be made. One evidence of this is the fact that an incomplete factorization requires fewer operations and less memory than a complete factorization, and, consequently, the time spent on complete factorization, for example by the PSLDLT solver, is the upper bound for the time needed for an incomplete factorization for the same problem.

For frequency sweep analyses the concept of using the same pre-conditioning, by means of an incomplete LDL^T factorization at one of the analysis frequencies, for analysis in a frequency band is evaluated. The results show that it is possible to improve the performance of the conjugate gradient method by this method. For perturbation analyses (e.g. to numerically evaluate gradients) this method may be even more attractive.

References

[1] **Axelsson O.**
Iterative Solution Methods.
Cambridge University Press, ISBN 0-521-44524-8 (1994).

[2] **Saad Y.**
Iterative Methods for Sparse Linear Systems.
PWS Publishing Company, ISBN 0-534-94776-X.

[3] **Hinton E. and Owen D.R.J**
Finite Element Programming
Academic Press 1977

[4] **Gustavsson M.**
Test Report on Saab 340 Test Section Ground Test.
ASANCA II Documentation, Deliverable 66/1B, Saab Aircraft AB, 1996.

[5] **Gustavsson M.**
Dynamic modeling of a turbo-prop aircraft using the U-vector Expansion Method.
Modern Practice in Stress and Vibration Analysis, Gilchrist (ed.) 1997 Balkema, Rotterdam.

[6] **Hörlin N, Tengzelius U.**
FRF:S from Analysis of Vibro-Acoustic FE Model of Trimmed Saab 340 Test Section
ASANCA II Documentation, Deliverable 66/2a, FFA Stockholm 1997.

[7] **Dovstam K.**
Augmented Hooke's law in frequency Domain.
A three dimensional, material damping formulation.
Int. Journal Solids and Structures, Vol. 32 No. 19, pp. 2835-2852, 1995.

[8] **Lesieutre, G. A.**
Finite Element Modeling of Frequency Dependent Material Damping using Augmenting Thermodynamic Fields.
Ph. D. Thesis, University of California, Los Angeles, 1989.

[9] **Taylor R.L, Beresford P.J & Wilson E. L**
A non-conforming element for stress analysis.
Int. Journal for num. methods in engineering, Vol. 10, 1211-1219 (1976).

References (continued)

[10] Gustavsson M, Hansson P.A
Dynamic Analysis of a shell using 'Morley' and 'enhanced strain' elements.
Div. of structural Mechanics (internal document), LTH, Lund University, Lund, Sweden 1998.

[11] Gustavsson M
Inverse Modeling of the dynamic properties of a turbo-prop Aircraft.
Div. of structural Mechanics, LTH, Lund University, Lund, Sweden 1998.

[12] Silicon Graphics
Manual pages for PSLDLT library psldlt.a