# Adaptive p-method in Pro/MECHANICASTRUCTURE TM

SP

Daf Special Products

# Adaptive p-Method in Pro/MECHANICA STRUCTURE™

**H.C. van Gompel**

**June 1996**

Daf Special Products
Technical report: TN-048

University of Technology Eindhoven
Faculty of Mechanical Engineering
Department WFW, practical report
WFW report number: 96-097

# SUMMARY

This study deals with a special method for the solution of finite element problems. This method, often referred to as p-version or p-method, is based on big, high order elements, in comparison to the traditional h-method, which is based on small, low order elements. The p-method has some advantages with respect to the h-method. The p-method is implemented in the software package Pro/MECHANICA STRUCTURE™.

The primary target of this research is to gain more insight into the solving routines used, the so-called adaptive p-method. The secondary target is to determine the influence of some important settings in the program on the quality and accuracy of the results, as well as on calculation times and disk storage. This is done in order to try to find settings to decrease the calculation times and disk storage.

Conclusions regarding the primary target are easily summarised. The theory behind the p-method, as it is implemented in STRUCTURE, is well described within this report. It was found that not all the advantages of the p-method, as described in literature, are used in STRUCTURE.

Finding a FE-model to determine the right settings with respect to the secondary target, was very difficult. With the models used in this study, the secondary target could not be fulfilled.

# TABLE OF CONTENTS

# 1. INTRODUCTION

Daf SP (Special Products) is active in the field of vehicle and aerospace systems. The major part of their operations is in the design and development of new products as well as production and maintenance primarily for the defence industry. Examples of production programs in operation are for the F-16 landing gear and the Leopard II tank, own development includes a light reconnaissance vehicle, and the landing gear and intermediate gearbox for the NH-90 helicopter.

Within the department T&O Aero of Daf SP the finite element program Pro/MECHANICA STRUCTURE™ (abb. STRUCTURE) is used. The program is part of a suite of programs which also includes the multibody program MOTION and the program for heat calculations THERMAL. There is a direct interface between all these programs and the corresponding CAD-program, Pro/ENGINEER. This interface and the set-up of STRUCTURE make it very user-friendly.

A finite element solution is always an approximation to the exact solution. When the accuracy of this approximation is asked for, the discretisation has to be refined and a new solution has to be calculated. This can be compared with the previous one and then it can be seen if the solution has converged or not. STRUCTURE uses the locally adaptive p-method to perform this, which is not as common as the 'standard' method, the h-method, used in most of the finite element programs. The h-method uses mesh refining where as the p-method uses higher order polynomials to describe the solution field.

Very long calculation times for big models make it sometimes necessary to abort an analysis before the final results are reached. STRUCTURE reports in a summary file the relative error reached, which sometimes after aborting, is still at 80%. When actual final values are determined, it is found that the results from the aborted run are within 5% of the final results. Such contradictions together with the long calculation times and the big temporary files, made it necessary to gain a better insight in the analysis methods used within STRUCTURE. The intention of the developer of STRUCTURE, is to make the program a design tool for engineers, instead of a tool for analysts. This also is a reason for gaining more insight to keep understanding the analysis methods in the more automised future versions.

In chapter 2 a general review of the finite element method is given and the different methods with their properties are discussed. The p-method is discussed more theoretically in chapter 3 where as chapter 4 deals with the different convergence algorithms as they are implemented in STRUCTURE. Practical models are tested in chapter 5 in order to extrapolate them towards bigger models and in chapter 6 the final conclusions and recommendations are given.

# 2. FINITE ELEMENT METHOD

## 2.1 General theory of FEM

The limitations of the human mind are such that it cannot grasp the behaviour of its complex surroundings and variations in one operation. Thus the process of subdividing all systems into their individual components or 'elements', whose behaviour is readily understood, and then rebuilding the original system from such components to study its behaviour is a natural way in which the engineer proceeds. This process is known as the element method. The elements used have a finite dimension and this property has led to the 'Finite Element Method', which is normally abbreviated to FEM [5].

For systems whose behaviour is not exactly known, mathematical models have to be formulated. These models are idealised representations of reality. When the mathematical model is very simple, an exact solution is possible. But most of the time the mathematical models are so complex that they do not permit exact solutions, thus out of necessity, the solutions are approximate. Correct interpretation of these approximate solutions is possible only if one is aware of the assumptions incorporated in the mathematical model and the associated limitations. This sequence of steps in the process of solving problems is illustrated schematically in figure 2.1.

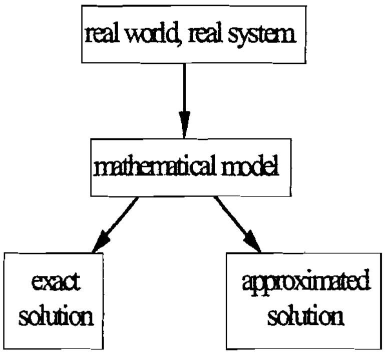


*figure 2.1: Solving scheme*

Because of the availability and cost of computational resources, mathematical models need to be simplified based on engineering experience and intuition. Most important is to determine which are the essential and nonessential factors with respect to the points of interest. For instance, in case of symmetry only half of the problem needs to be taken in account, with a set of accompanying boundary conditions of course.

The finite element method should be understood as a method for finding an approximate solution for a simplified model. Numerical treatment reduces the simplified model to a form which is solvable by a finite number of numerical operations. This means that the approximate solution has to be characterised by a finite number of parameters, called degrees of freedom. Such a reduction of the problem, so that representation by a finite number of parameters is possible, is called discretisation.

## 2.2 Different methods in FEM

The theory of the different methods in FEM are described according to the definitions in Szabó & Babuška [4].

The finite element method selects a solution $u_{FE}$ from the finite element space of admissible functions $\tilde{S}$ which minimises the energy norm or the error:

$$\left\| u_{EX} - u_{FE} \right\|_{E(\Omega)} = \min_{u \in \tilde{S}} \left\| u_{EX} - u_{FE} \right\|_{E(\Omega)} \tag{1}$$

in which $E(\Omega)$ is the set of all functions which have finite strain energy on the domain ($\Omega$) and is called energy space. E($\Omega$) endowed with the norm $\left\| \cdot \right\|_{E(\Omega)}$ is a normed linear space as described in reference [4].

This relationship indicates that the error depends on the exact solution $u_{EX}$ and the space $\tilde{S}$ which is determined by the mesh, the polynomial degrees of elements, and the mapping functions, collectively called the already mentioned discretisation. Discretisation is controlled by the users of finite element computer programs, either directly or through procedures designed to select or modify certain discretisation parameters automatically on the basis of data generated in the course of the solution process.

Engineering computations are performed for the purpose of obtaining information concerning the expected response of physical systems to certain imposed conditions, generally called loads. This information is then used in making engineering decisions. Obviously, the computed data must be of such quality that decisions based on them will be substantially the same as if the exact solution were known. Therefore, we wish to select the discretisation so that $u_{FE}$ is close to $u_{EX}$, in some sense.

In general, we wish to determine functionals $\Psi_i(u_{FE})$ (i = 1, 2, ...), such as displacements, stresses, reactions, stress intensity factors, etc.. so that:

$$\left| \frac{\Psi_i(u_{EX}) - \Psi_i(u_{FE})}{\Psi_i(u_{EX})} \right| \le \tau_i \qquad (i = 1,2,\ldots) \tag{2}$$

where $\tau_i$ represents a specific tolerance.

The numerical solution for the functionals, $\Psi_i(u_{FE})$, has to be close to the exact solution for the functionals, $\Psi_i(u_{EX})$. It is difficult to say how close these two are because we don't know the exact solution $u_{EX}$. The way to determine this is by performing extensions and certain tests on the finite element solutions. Both the estimation and control of the errors of discretisation are based on extensions.

### 2.2.1 *Extensions*

Extensions are systematic changes of discretisation so that the number of degrees of freedom is increased at each change. More precisely, a sequence of finite element spaces $S_1$, $S_2$, $S_3$ ... with progressively improved approximation properties is created and the corresponding solutions obtained. If extension is based on mesh refinement, then the process is called h-method. If extension is based on increasing the polynomial degree of elements, then the process is called p-method. If extension is based on a combination of concurrent mesh refinement and increase in the polynomial degree of elements, then it is called hp-method. Extensions provide information on the basis of which we can draw conclusions concerning the overall quality of the finite element solution $u_{FE}$ and the accuracy of functionals computed from $u_{FE}$. When convergence of the finite element solutions corresponding to spaces $S_1$, $S_2$, ... is of interest, then we refer to h-, p-, or hp-convergence.

An extension can be submitted manually or automatically. It is called adaptive if the extension is submitted automatically. When a specific convergence is wanted, a convergence criteria is needed to test if that convergence is achieved or not. If convergence is not achieved, the extension will be submitted and this solution will be tested again, using the convergence criteria. Examples of such criteria, as they are used in STRUCTURE, are edge displacement and element strain energy.

With edge displacement for convergence, the functionals $\Psi_i(u_{FE})$ in equation (2) will be edge displacements. Equation (2) has to be fulfilled for displacement along every edge before convergence is achieved.

If element strain energy is used as a convergence criteria, the functionals $\Psi_i(u_{FE})$ in equation (2) are the strain energies of each element independently. A definition for strain energy is [7]:

> *'Mechanical energy stored up in stressed material. Stress within the elastic limit is implied; therefore, the strain energy is equal to the work done by the external forces in producing the stress and is 100% recoverable.'*

A basic formula to calculate the strain energy of an element is in equation (3) taken from reference [4],

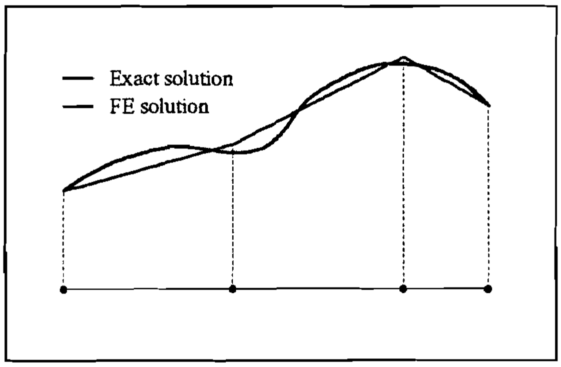
$$\int_V \delta\, \varepsilon^T \sigma \, \mathrm{d}V \tag{3}$$

in which $\delta\varepsilon$ are the variations of the strains, $\sigma$ are the stresses and $V$ is the domain of the element.

### 2.2.1.1 h-method

Most of the finite element codes rely on the h-method, in which a deformable mechanical structure is analysed by subdividing it into elements within which the field variables are approximated by low order polynomial functions, often linear, and sometimes quadratic or cubic. The accuracy of the analysis is increased by refining the element mesh, decreasing the size of the elements (h).

An example with three linear elements is shown in figure 2.2. The exact solution is visualised by the solid line and the approximating finite element solution by the dashed line. In figure 2.3 the same problem with a finer mesh of five linear elements is shown. It is clearly visible that the mesh refinement makes it possible to approximate the exact solution more accurate.



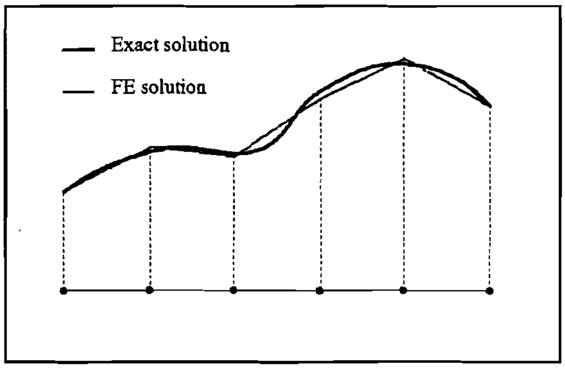


*figure 2.2: Coarse mesh, h-method*

*figure 2.3: Fine mesh, h-method*

The FE-solutions shown in figure 2.2 and figure 2.3 are determined with a variational method and not with a collocation method. With the variational method the energy of the solution is minimised over an element. Therefore the solutions on the nodal points are not necessarily the exact solutions. With a collocation method the solution is only calculated on the nodal points.

### 2.2.1.2 p-method

In the p-method, the basis functions for representing the field variables are high-order polynomials, and convergence towards an exact solution is obtained by increasing the order of approximation within each element, increasing the order of the polynomial functions (*p*). Mesh refinement is therefore not required for convergence.

When the examples of figure 2.2 and figure 2.3 again are used, the p-method uses for its first approximation the same mesh with the three linear elements. When the p-order is increased in order to get a better approximation the coarse mesh will still be the same, but the approximation between the nodal points can be described with an quadratic solution field instead of a linear. So the FE-solution will consist of three quadratic polynomials between the nodal points and it is obvious that this will describe the exact solution better than the linear FE-solution. In order to make a comparison between the h-, and p-method an example with a simply supported beam is shown in figure 2.3.

The analytical solution for the deflection in the y-direction alongside of the beam is taken from literature [7]. It is a third order polynomial function, see equation (4).

$$y(x) = \frac{-F}{3 \cdot E \cdot I} \cdot x^3 \tag{4}$$

When only one linear element is used with the h-method, it is obvious that this deflection can not be calculated exactly. When the size of the elements is decreased, more elements will be generated alongside of the beam. The FE-solution will converge in the direction of the analytical solution with this mesh refinement. With the p-method one linear element can not describe the deflection accurate, this is the same as with the h-method and one element. When the polynomial order is increased, the FE-solution will approximate the analytical solution better. With polynomial order three, the analytical solution will be the same as the exact solution.

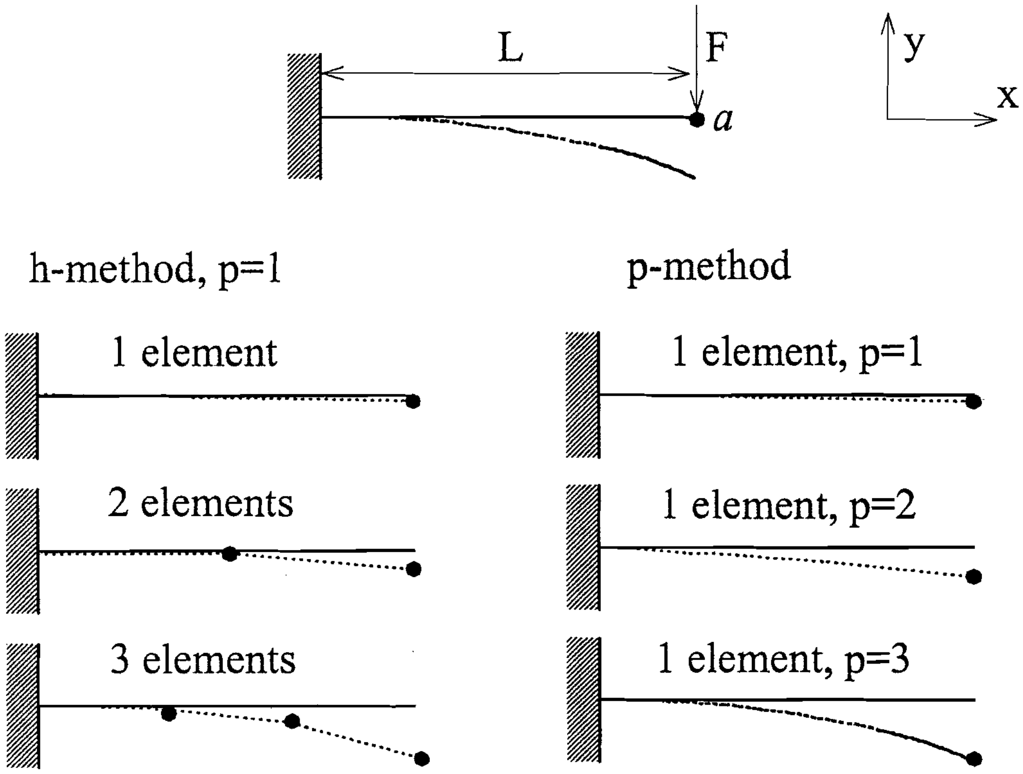


*figure 2.4: Representation of the h-, and p-method with a simply supported beam*

In figure 2.4 it is shown that a good solution, in this case the exact solution, can be obtained with the p-method, using only one element. This illustrates the large elements which are possible with the p-method. The possibility of large elements in the p-method is also available in STRUCTURE. The default value for the aspect ratio within STRUCTURE for instance is 30, where as in the h-method maximum aspect ratios of 6 are advised.

Further details of the p-method will be discussed extensively in chapter 3.

### 2.2.1.3 hp-method

As the name already suggests, the hp-method is a mixture of the h-, and p-method. It is a combination of concurrent mesh refinement and increase in the polynomial degree of elements. This method is proven to converge faster than the h-, or p-method. The hp-method is not yet available in commercial FE-codes.

## *2.2.2 Convergence*

The convergence rates of the three different methods (h-, p-, and hp-method) are very different. It has been proven that the convergence of the h-method is slower than the p-method and that the p-method again is slower than the hp-method [3]. To illustrate the difference in rate of convergence schematically, the relative error of a random criteria is plotted against the number of degrees of freedom for the three different methods in figure 2.5. This particular form of the figure would occur for the relative error of every other criteria as well.

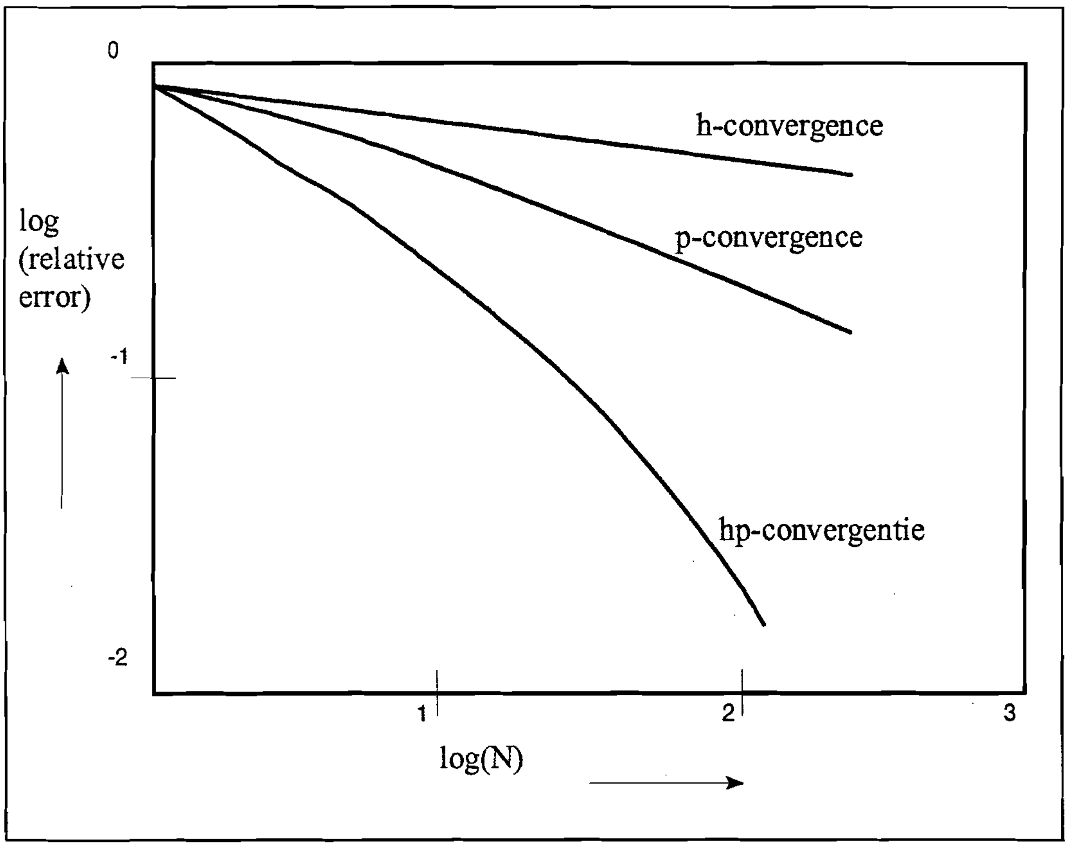


*figure 2.5: Schematic illustration of the convergence rate of h-, p-, and hp-method*

In figure 2.5 the convergence rates of the three different methods are clearly visible.

# 3. THE P-METHOD IN STRUCTURE

## 3.1 General theory

Pro/MECHANICA STRUCTURE™ is based on the p-version of the finite element method, or p-method, which has been treated in the research literature for about 25 years. The first theoretical discussions were done by Zienkiewicz and Taylor [5]. Further theoretical foundations of the p-method are discussed by Babuška, Szabó, and Katz [2], and Babuška, and Szabó [3].

In the p-method, the basis functions for representing the field variables are high-order polynomials, and convergence is obtained by increasing the order of approximation within each element. Mesh refinement is therefore not required for convergence.

In practice, it is not possible to increase the order of the elements indefinitely. The elements in STRUCTURE use polynomial basis functions up to ninth order, which is sufficient for accurate representation of the physical behaviour of most structures in regions where displacements and stresses are smoothly varying. In the neighbourhood of cracks, re-entrant corners, and concentrated loads, however, stresses can be singular (theoretically infinite); elements must be further subdivided near such local zones in order to obtain more accurate results.

## 3.2 Implementation of the p-Method in STRUCTURE

STRUCTURE uses a displacement formulation, in which stiffness matrices and load vectors are derived by minimising the structure's potential energy expressed in terms of displacements and rotations. As in other finite element programs, displacements within each element are represented by a linear combination of basis functions N, known as shape functions, multiplied with coefficients $a_i$, see equation (5).

$$u = a_1 N_1 + a_2 N_2 + \ldots + a_n N_n \qquad (5)$$

In the h-method the coefficients are nodal displacements. In the p-method, however, the primary variables are not nodal displacements, but are coefficients of linear, quadratic, cubic, and higher order polynomial basis functions, up to order nine in STRUCTURE. Once the finite element equations are solved for these coefficients, physical displacement and stress results are calculated over a grid as a post-processing step. During the pre-processing a value for the plotting grid has to be given. This number for the plotting grid is the number of points on which the physical values have to be calculated by solving an equation as in (5).

The shape functions are hierarchical and are based on the integrals of the Legendre polynomials. Hierarchical shape functions are a special kind of shape function which give certain advantages over the 'standard' shape functions. The hierarchical and standard shape functions will be discussed in later sections. Shape functions based on the integrals of Legendre polynomials are recommended by Babuška, Szabó, and Katz [2] because they have some orthogonality properties which result in well conditioned stiffness matrices.

The shape functions are generated by taking products of the polynomials in the different natural coordinate directions, and can be separated into four types:

1. Linear nodal functions (vertex modes), which vanish at all but one node in an element. These are identical to the conventional shape functions in h-elements.
2. Edge functions (side modes), which are non-zero along a particular element edge and adjacent element faces and volumes.
3. Face functions (internal modes), which are non-zero along a particular element face and adjacent element volumes.
4. Volume functions, which are non-zero within a solid element, but vanish on the faces.

To visualise the different types of functions, in figure 3.1 [4] the hierarchical shape functions for quadrilateral elements are shown. A quadrilateral element is a two dimensional element so only the vertex, side, and the internal modes are illustrated. For the volume functions the graphical representation is only possible with three dimensional elements and is somewhat more difficult, so this is left to the imagination of the reader.

Block A contains the vertex modes for the four nodes. The numbers 1 to 4 above block A are the numbers of the node at which the shape function below belongs. In block B the side modes are represented. The numbers on the left are the polynomial orders of the shape functions (*p*). Block C shows the internal modes for the face of the quadrilateral element. It is clearly visible that for the higher order shape functions (block B and C) there still only are four nodal points. For the higher order functions no extra nodal points have to be inserted. This is a specific property for the hierarchical shape functions and will be explained in a later section.

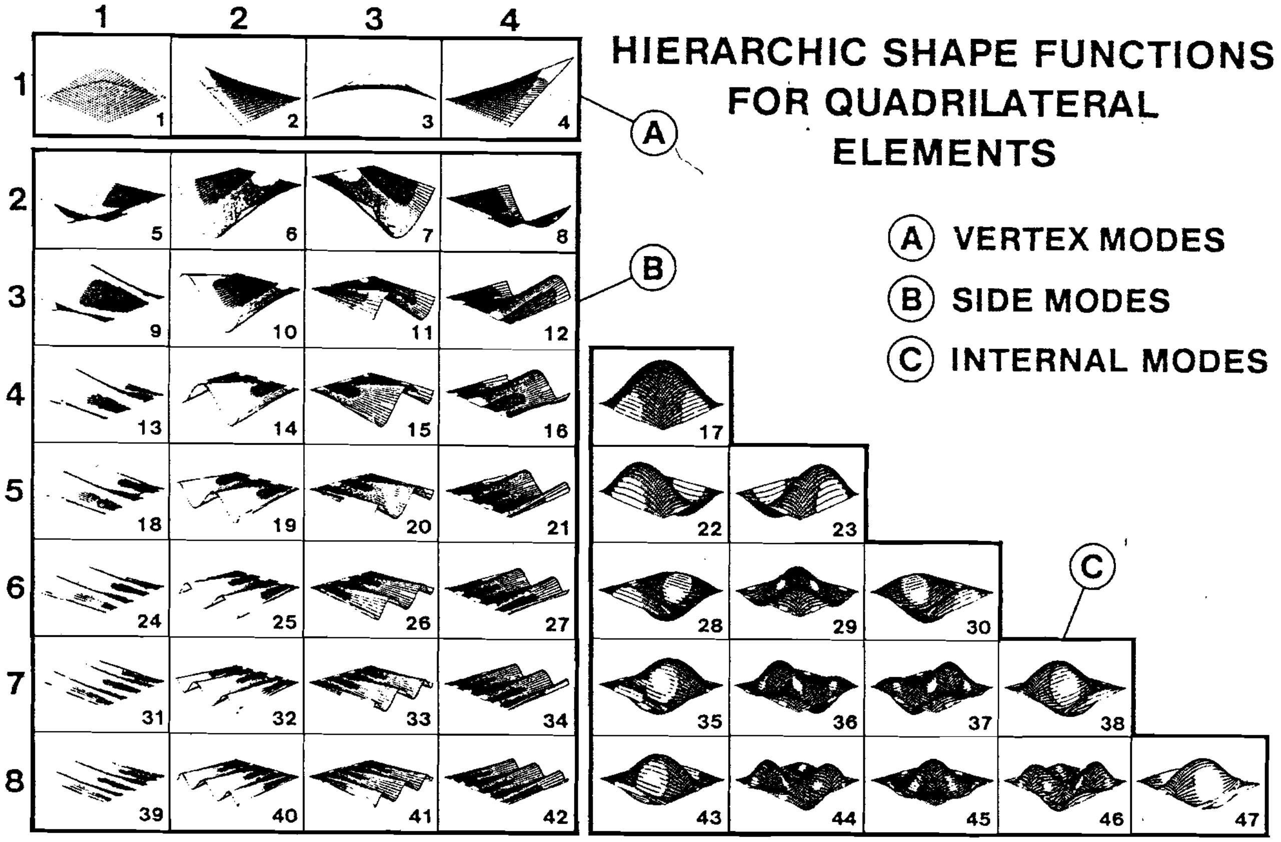


*figure 3.1: Hierarchic shape functions for quadrilateral elements*

Conventional h-element formulations usually rely exclusively on functions of type 1 above, the vertex modes. Occasionally second order functions of type 2, side modes, are employed to transition between shell elements of different order. Second order functions of type 3 or 4, internal modes or volume functions, are sometimes included as "bubble" modes for shells of solids.

## 3.3 Element formulation

The displacement and rotation field variables are expressed in terms of polynomial series:

$$u = \sum_{i=1}^{n} \begin{bmatrix} u_x^i \\ u_y^i \\ u_z^i \end{bmatrix} N^{(i)} \qquad \beta = \sum_{i=1}^{n} \begin{bmatrix} \beta_x{}^i \\ \beta_y{}^i \\ \beta_z{}^i \end{bmatrix} N^{(i)} \qquad (6)$$

where $u$ and $\beta$ are the displacement and rotation vectors, the functions $N^{(i)}$ are the hierarchical polynomial basis functions given by [2], $u_x^i$, $u_y^i$, $u_z^i$, $\beta_x^i$, $\beta_y^i$, $\beta_z^i$ are the coefficients in the expansion for the Cartesian components, and $n$ is the number of terms in the series, which include polynomials up to order p. For the implementation of the elements, polynomial basis functions up to ninth order ($p$=9) are used. Further details regarding the formulations for the different elements are given in chapter 3 of [1].

## 3.4 Shape functions

The essence of the finite element method is in approximating the unknown variable field by an expansion like in the previous paragraph, see equation (5). This for a scalar variable ($u$) can be written as in equation (7).

$$u \approx \hat{u} = \sum_{i=1}^{n} N_i \, a_i \tag{7}$$

where $a_i$ are the unknown parameters to be determined, the previously mentioned coefficients, $\hat{u}$ is the FE-approximation for the scalar variable $u$ and $N_i$ are the shape functions. For each element shape functions are generated for each node independently, and are described with so called iso-parametric coordinates. This is illustrated in figure 3.2 for a two dimensional element. Shape functions should be defined in such a way that the value of the function is one in the corresponding nodal point and zero in every other nodal points. The exact description of the functions is free to chose.

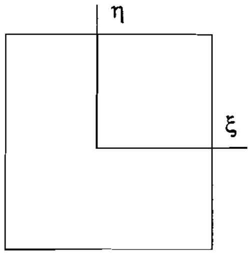


*figure 3.2: 2-D iso-parametric coordinates*

It is possible to choose to identify the coefficients $a_i$ with the values of the unknown function at element nodes, the nodal value becomes the coefficient value, thus making

$$u_i = a_i \tag{8}$$

When you are for instance looking at displacements as the unknown variable field, the coefficients are the nodal displacements. The shape functions so defined that equation (8) is valid, will be referred to as the already mentioned 'standard' ones and are the basis of most finite element programs.

When element refinement is made with standard shape functions, extra nodal points are placed along the element edges and totally new shape functions have to be generated for these extra nodal points. Hence all calculations have to be repeated.

Element refinement with hierarchic shape functions does not produce new nodal points. This means that the old set of shape functions can be used again when a polynomial order update is submitted. The coefficients belonging to the new shape functions no longer have an obvious physical meaning but are mathematical coefficients, equation (8) is not valid in this case. The hierarchic and standard shape functions are schematically illustrated in figure 3.3 by a one dimensional element.

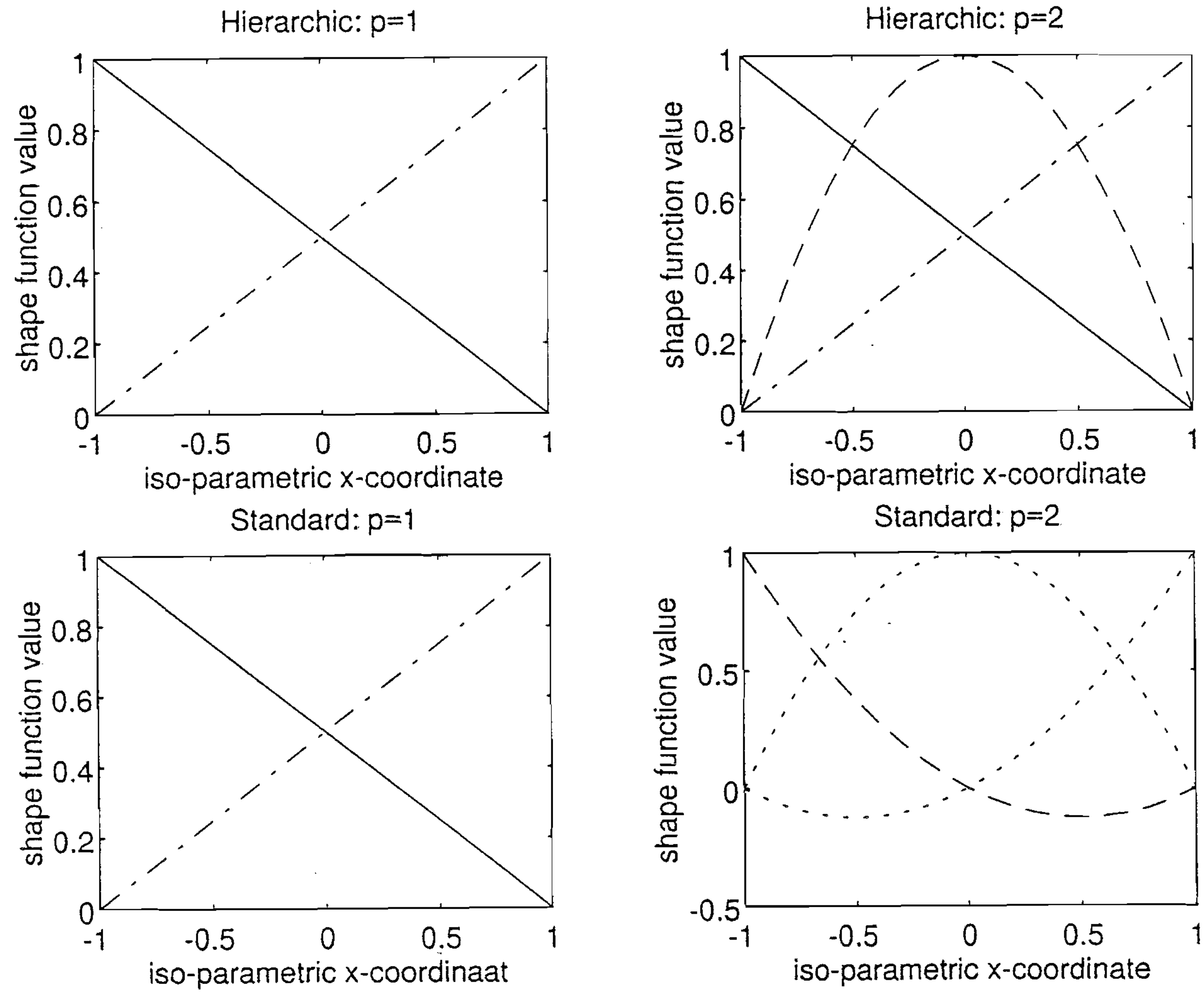


*figure 3.3: Schematic representation of the standard and hierarchic shape functions for a one dimensional element*

The four plots show the hierarchic and standard shape functions for a first and second order description. On the horizontal axis the iso-parametric x-coordinate is shown and on the vertical axis the value of the shape functions. For the hierarchic shape functions it is clearly visible that the two linear shape function (p=1), also appear when a quadratic description (p=2) for the element is used. A quadratic function also appears, but it is not actually connected to a nodal point. With the standard shape functions the two linear functions disappear when the quadratic form is used. This is obvious because an extra nodal point is placed in the middle of the element, $\xi=0$. At this nodal point the shape functions for the points $\xi=-1$ and $\xi=1$ have to be zero, so the old linear shape is false.

Especially the building of the stiffness matrix in a finite element program is a time consuming activity. Every time new shape functions are generated, that matrix has to be calculated again. It would be of advantage to avoid this difficulty by considering equation (7) as a series in which the shape functions ($N_i$) do not depend on the number of nodes in the mesh ($n$). This indeed is achieved with hierarchic shape functions.

## 3.5 Hierarchic shape functions

As discussed in the previous section the unknown variable field can be approximated with an expansion like in equation (7). Already it was mentioned that two types of shape functions are possible, the standard and the hierarchic. The advantage in using hierarchic shape functions is in the fact that the old set of functions can be used in a new set after a polynomial update has been done. If the variables are properly numbered, then the stiffness matrix (resp. load vector) corresponding to polynomial order $p$ is embedded in the stiffness matrix (resp. load vector) corresponding to order $p$+1, which again is embedded in the stiffness matrix (resp. load vector) corresponding to order $p$+2, etc. This is schematically illustrated in figure 3.4 [4].

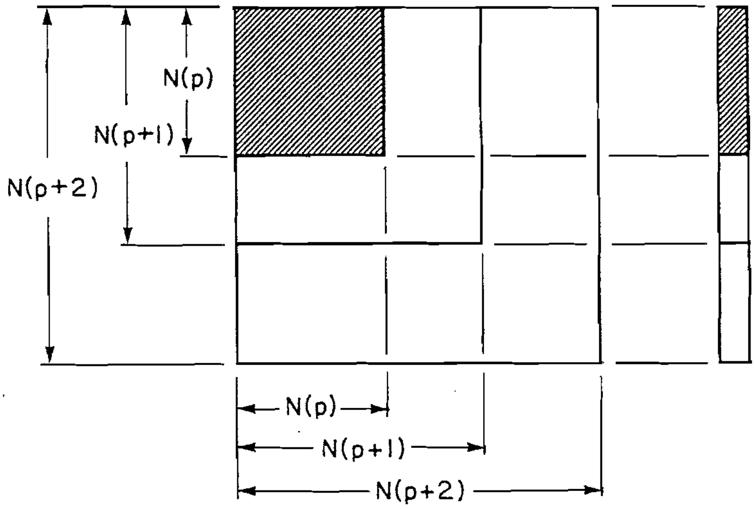


*figure 3.4: Schematic representation of the hierarchic structure of stiffness matrices and load vectors in p-extension*

The hierarchic structure of stiffness matrices and load vectors can be exploited in the solution phase. For instance: having performed Gaussian elimination and obtained a finite element solution corresponding to order $p$, that solution can be used again and elimination can be continued to obtain the finite element solution corresponding to order $p$+1.

When a higher order shape function is added to the old set of shape functions, the stiffness matrix and load vector will be expanded as is schematically illustrated in figure 3.4. These new matrix and vector give a new set of equations. Solving this new set of equations means that the already calculated lower order coefficients will change. In equation (9) a possible set of equations is given.

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{9}$$

The first equation from this set is in equation (10).

$$k_{11}a_1 + k_{12}a_2 + k_{13}a_3 + k_{14}a_4 = f_1 \tag{10}$$

When the set of shape functions is expanded, equation (9) will be expanded as well. The first equation from this new set of equations will become:

$$k_{11}a_1^* + k_{12}a_2^* + k_{13}a_3^* + k_{14}a_4^* + k_{15}a_5^* + \ldots + k_{1n}a_n^* = f_1 \qquad (11)$$

When equation (11) is solved, the coefficients $a_1^*$ until $a_4^*$ will be different from the coefficients $a_1$ until $a_4$, which can be solved from equation (10).

The first two coefficients $a_1$ and $a_2$ are belonging to the linear shape functions and are equal to the nodal values, for whatever functional is solved. With the variational method used here, it is possible that when these first two coefficients are calculated, they do not represent the exact nodal value, see figure 2.2. The value of the already calculated coefficients will change when the polynomial order is increased; when the set of shape functions is expanded, as previously discussed. This makes it possible that the first two coefficients can represent the exact nodal values better.

## 3.6 Hierarchic polynomials in one dimension

In the previous sections the general ideas of hierarchic approximation are introduced. The idea of generating higher order hierarchic forms is simple. To show this, a one-dimensional expansion is given in this paragraph. This provides a basis for generation of two- and three-dimensional forms.

To generate a polynomial of order *p* along an element side we do not need to introduce nodes but can instead use parameters without an obvious physical meaning, known in dynamics as a Ritz approach. We could use here a linear expansion specified by the two linear 'standard' functions $N_0$ and $N_1$ and add to this a series of polynomials always so designed as to have zero values at the ends of the range (i.e., points 0 and 1). Hierarchical shape functions ($N_2 \ldots N_n$) always have a value of zero at the ends of the element edge in order to make the compatibility between different elements accurate, see reference [4]. In figure 3.5 [5] the different hierarchical shape functions are illustrated.

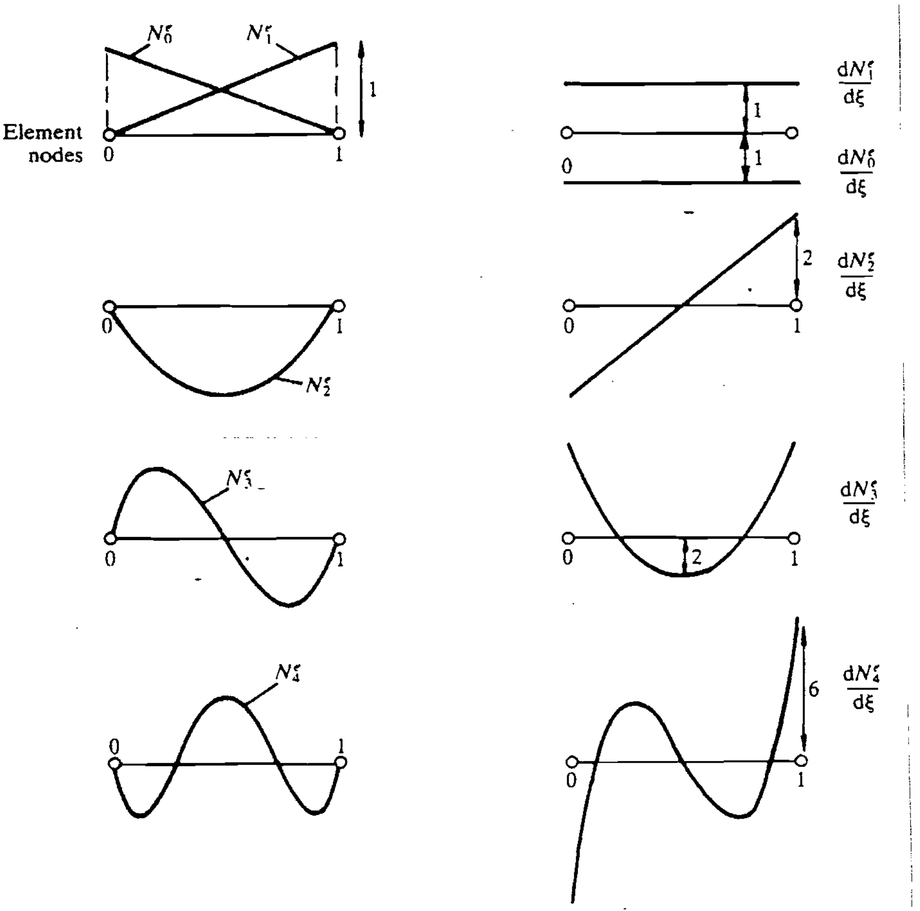


*figure 3.5: Hierarchical element shape functions of nearly orthogonal form and their derivatives*

Thus for a cubic approximation, over the typical one-dimensional element, the approximating function would be the linear expansion in the following equation (12):

$$\bar{u} = u_0 N_0 + u_1 N_1 + a_2 N_2 + a_3 N_3 \tag{12}$$

The linear functions $N_0$ and $N_1$ are the same for whatever the shape functions are. The higher order shape functions can be defined in various ways as long as the value of the functions is zero at the points 0 and 1 ($\xi$=-1 and $\xi$=1). The coefficients $u_0$ and $u_1$ are the nodal values and the coefficients $a_2$ and $a_3$ are mathematical coefficients.

The first two coefficients Since STRUCTURE is working with a variational method, the first

## 3.7 Optimal form of the shape functions

An optimal form of the hierarchic functions is one that results in a diagonal equation system. Such an equation system means that the off-diagonal terms in an element stiffness matrix should be zero. For such a system orthogonality in the shape functions is needed. With orthogonality the coupling between successive solutions disappears. A diagonal equation system is easier to solve than a system with full matrices.

Within STRUCTURE the integrals of Legendre polynomials are used as shape functions. The set of Legendre polynomials $P_p(\xi)$ possesses the orthogonality property over the range $-1 \leq \xi \leq 1$, where $\xi$ is the iso-parametric x-coordinate along the element edge in an one dimensional problem as in figure 3.5. Also the integrals of the Legendre polynomials posses this property.

The reason for choosing the integrals of the Legendre polynomials and not the Legendre polynomials themselves, was not found during this research. Some important properties of Legendre polynomials are listed in appendix A.

The integrals of the Legendre polynomial, the shape functions in STRUCTURE, of degree $p$, are defined by equation (11):

$$N_{p+1}^{e} = \int P_p(\xi)d\xi = \frac{1}{(p-1)!2^{p-1}}\frac{d^{p-1}}{d\xi^{p-1}}\left[(\xi^2-1)^p\right] \tag{11}$$

As mentioned before the shape functions $N_0$ and $N_1$ are the same as with every other choice for the shape functions

$$N_0^e = -\frac{\xi-1}{2} \qquad N_1^e = \frac{\xi+1}{2} \tag{12}$$

Evaluation of equation (12) for each $p$ in turn gives:

$$N_2^e = \xi^2-1 \qquad N_3^e = 2(\xi^3-\xi) \quad \text{etc.} \tag{13}$$

The shape functions plotted in figure 3.5 are the integrals of the Legendre polynomials as they are used in STRUCTURE.

A completely diagonal equation system is obviously not possible, the different coefficients would not be connected and so the different element would not be connected. A matrix with a diagonal band is possible and this also leads to numerical advantages over a fully filled matrix.

From literature [5] it is found that hierarchical shape functions should lead to a more diagonal system as with standard shape functions. However, when large models were analysed with STRUCTURE, it was found that the bandwidth of the stiffness matrix was relatively small in the beginning of an analysis (low $p$-levels). For higher polynomial orders the maximum bandwidth becomes nearly the same as the number of equations, the width of the matrix itself. The average bandwidth stays considerably small in comparison with the number of equations, indicating that the maximum bandwidth is caused by only a few terms in the matrix which are spread throughout the matrix. What causes this large spreading of the terms, is not found.

After a model is analysed, several properties of the stiffness matrices are summarised in a output file. For instance the maximum bandwidth and the average bandwidth. After checking these files for a few large models as they were analysed within SP, it seems that the speed of solving the equation systems with STRUCTURE is dependent of the maximum bandwidth and not the average bandwidth. This probably is a reason for the large calculation times. According to information from the supplier of STRUCTURE, there will be a sparse solver in future releases. This sparse solver should detect the zero terms in the matrix and solve the equations more efficiently. The maximum bandwidth will be of less importance then.

# 4. CONVERGENCE ALGORITHMS

## 4.1 Different algorithms

Within STRUCTURE two convergence algorithms are possible: multi-pass and single-pass. With multi-pass the solution is obtained by successively increasing the polynomial order from the minimum specified order until convergence or the maximum order is reached. Single-pass only does two calculations in which the FE analysis is solved. The first calculation is solved with a fixed polynomial order and then an error is determined with which the polynomial order update is done. The second calculation is solved with this updated model. The descriptions of these two algorithms are described in the next two sections.

## 4.2 Multi-pass

Within multi-pass STRUCTURE uses a locally adaptive polynomial order update scheme. In this scheme the updating of the polynomial order is done automatically (adaptive) and a polynomial order, one through nine, can be assigned independently to each element edge (locally). The convergence algorithm is based on an iterative procedure in which the analysis is solved for successively higher polynomial order. The user can choose a criteria for which the convergence will be checked. This criteria can be all sort of quantities, for instance displacements, stresses or moments. If the convergence check shows that the demanded convergence was not reached, the polynomial order has to be updated. The algorithm which drives the polynomial order update is based only on edge displacements and element strain energies, according to reference [1]. The criteria edge displacements and element strain energies are already discussed in chapter 2.

### *4.2.1 Convergence check*

To check the convergence, STRUCTURE offers several possibilities. There are two standard convergence criteria and there is the possibility to choose a self defined measure, for instance stress, displacement or moment. The two standard criteria are:

1. Local displacement & local strain energy

   When local displacement & local strain energy (LD&SE) is used for convergence, STRUCTURE calculates until convergence is reached for the two convergence criteria, for the displacements along each edge and for the strain energy of each element. According to the manual [8], this option should be used if the overall displacement solution is important. Stresses however, may not converge using this option.

2. Local displacement & local strain energy & RMS stress

   With this option, STRUCTURE calculates until reaching convergence for the RMS stress for the entire model, in addition to edge displacement and strain energy. According to the manual [8], STRUCTURE uses a more conservative method for converging on RMS stress in comparison to the previous option. This implicates that this choice for convergence criteria, is likely to yield accurate results for most quantities, although it also increases the computation time for most runs.

### *4.2.2 Multi-pass convergence algorithm*

The convergence algorithm consists of the following steps [1]:

1. At the start of the calculation the polynomial order of all edges is set to the minimum polynomial order specified by the user. The default minimum polynomial order is 1.

2. A finite element solution is carried out.

3. Point displacements and rotations over a grid of sample points along each element edges are calculated and stored. Also the total strain energy of each element is calculated and stored.

4. If this is the first iteration, all the edges are updated by one polynomial order, and steps (2) and (3) are repeated. A minimum of two successive iterations is required in order to proceed with the convergence check devised in the subsequent steps.

5. The total strain energy of the model is calculated and stored, and the error in the global energy norm (an RMS stress measure) is estimated from the three point extrapolation formula given by Szabó and Babuška [4]. If this is the second iteration, the error is estimated from the strain energy change relative to the first iteration, without extrapolating. This calculated quantity is only used for convergence checking when 'Local displacement & local strain energy & RMS stress' is used as a convergence criteria.

6. The local edge displacements and element strain energies are compared with the values from the previous iteration. When local displacement and strain energy are selected for convergence, the local displacement and strain energy percent differences have to be less than the user input tolerance, to consider the results to be converged and stop the iteration. When the RMS stress is taken in account for convergence as well, the percent difference in estimated error in the global energy norm also has to be less than the user input tolerance. The iteration also stops if the user has selected a particular response quantity for convergence control, a measure, and the change in that quantity is less than the tolerance.

7. If the percent differences between the current results and the results of the previous iteration are greater than the user input tolerance, the polynomial orders of the edges are updated.

8. The following rules determine which edges are updated:

   a) Edges for which the differences in displacements exceed the tolerance are updated by one polynomial order. If the coefficients of all odd or all even symmetry functions along a particular edge are negligible, the edge is updated by two polynomial orders to account for symmetry. The coefficients of the higher-order shape functions are also examined. If it is found that the higher-order coefficients are small relative to the lower-order coefficients, the update is not performed. This prevents unnecessary updates due to rigid motion.

   b) If the square root of the difference in strain energy of an element relative to the previous iteration exceeds the tolerance, all edges of the element which have not already been updated in conjunction with (a) are updated by one polynomial order. The energy content of the higher-order shape functions is also examined. If it is found that the higher-order functions contribute negligibly relative to the lower-order functions, the update is not performed. This prevents updates due to changing strain energy contributions of lower-order shape functions.

c) For multiple load sets or modes, the update patterns for each load set or mode are merged with an 'or' option. The final update pattern therefore reflects the combined effects of all the load sets or modes. Different update patterns, and hence slightly different results, are obtained if two load sets are run in the same analysis or in two separate analyses.

9. If the polynomial order update requires that the polynomial order of an edge exceed the maximum allowable order, the results are considered to be not converged and the iteration stops. Otherwise, another 'p-loop' iteration is carried out by returning to step (2) and performing a new analysis with the updated polynomial orders.

In the above iterative procedure, it is possible that the displacements will converge more rapidly along some edges than others, and that the strain energies of some elements will converge more rapidly than others. Once convergence is reached for an edge or element, the local polynomial order update stops, although the polynomial order may continue to be updated for edges in other parts of the model where stress and displacement gradients are higher. This often leads to an inhomogeneous distribution of edge orders, which is computationally efficient, since high-order shape functions are employed only where they are needed.

The above convergence algorithm estimates error by the size of 'the last term in the series', and does not predict the exact error. Also, the convergence algorithm is driven by displacements and strain energy, which are known to converge faster than stresses. For stress convergence, a stress quantity should be selected as a response quantity to control convergence. To interpret the quality of the results, it is possible to examine convergence curves for specific quantities of interest. The value of the quantity is then plotted versus the pass number, see figure 4.1.

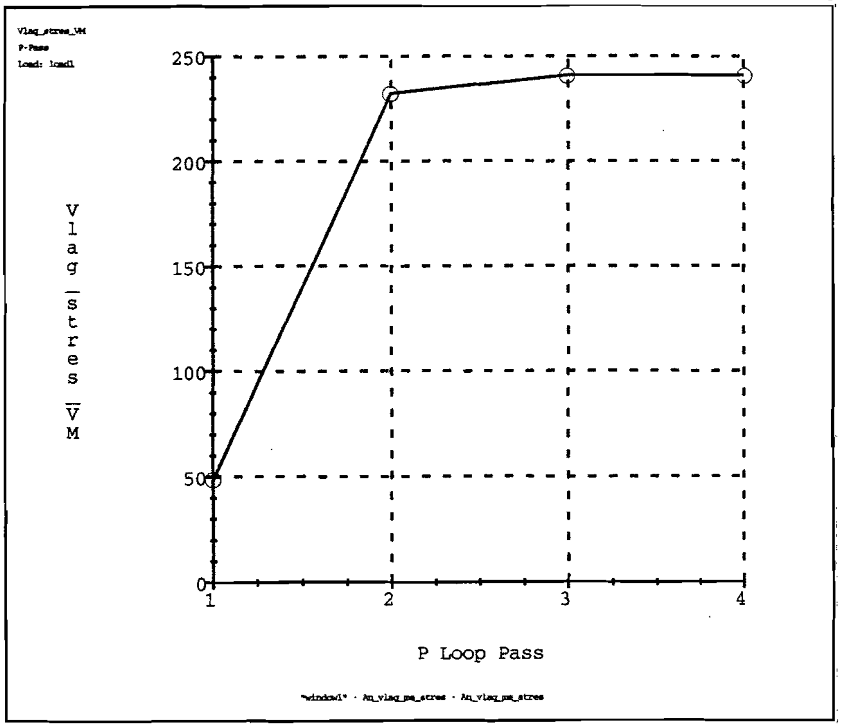


*figure 4.1: Convergence plot*

In figure 4.1 the value of a measure on one point in a model vs. the pass number of an analysis is plotted. The used measure in figure 4.1 is Von Mises stress. The relative difference between successive values should be smaller than a user defined tolerance. In this example the relative difference between the value of pass 3 and the value of pass 4 is smaller than the specified tolerance, so the iteration stopped after pass 4.

## 4.3 Single-pass

### *4.3.1 General theory of single-pass*

For the single-pass adaptive convergence algorithm, the engine first solves the problem at polynomial order 3, estimates stress errors, raises the polynomial orders of each element to a higher level based on the magnitude of the local stress errors, and then carries out a second solution using the updated polynomial orders. The results of this second solution are output as the final results.

After solving with the updated polynomial orders, the engine calculates a stress error measure and writes it to the summary file. This provides feedback as to the level of accuracy of the solution, similar to the convergence data supplied by the multi-pass approach.

According to the European customer support, the new algorithm is expected to deliver results which are accurate for most engineering purposes, i.e., a comparable level of accuracy to the multi-pass algorithm with default settings, but with much better perfomance than the multi-pass algorithm for most models, both in terms of time and disk/memory resources. Although the new algorithm does not rely on a convergence loop and hence provides no convergence plots, the new algorithm is adaptive, and is well differentiated with respect to accuracy relative to h-element discretisation.

If the user elects to use the iterative solver, the engine first solves the problem at polynomial order 2 with the block solver, then uses the iterative solver for the uniform polynomial order 3 pass and for the third pass where the edges have been updated to their final values. The polynomial order 2 pass with the block solver is necessary for the engine to compute an efficient preconditioner for the iterative solver.

In STRUCTURE version 8 the use of single-pass is restricted to solid models with a single isotropic material. In future releases single-pass will be available for a wider range of models.

### 4.3.2 Error estimates

The stress error estimates reported in the summary file are calculated by sampling the local error estimates (the same used to update polynomial orders with single-pass). The local stress error estimates for the new algorithm are obtained by comparing the superconverged stress values with the conventionally calculated stress values. This method of improving stress accuracy and estimating stress error is based on the work by Zienkiewicz and Zhu [6], sometimes referred to as the $Z^2$-method, for further information the reader is referred to [6].

Superconvergent stress recovery was already implemented in STRUCTURE version 7.1 to improve stress accuracy, and in STRUCTURE version 8 the same technology is employed as the basis for the single-pass adaptive convergence algorithm.

### 4.3.3 Use of new algorithm in practice

According to the European customer support, internal tests of the single-pass adaptive algorithm on typical models in the several thousand element range have shown that results are within 1% on displacement and 5% on stress, relative to the multi-pass algorithm with default settings. Those default settings are local displacement and local strain energy with a tolerance of 10%. For the block solver, they have observed that on average, the engine runs 10 times faster, and uses 1/4 as much disk space with the single-pass algorithm than with the multi-pass algorithm with default settings. For the iterative solver, the corresponding figures are 3 times faster with 2/3 disk space usage, in comparison to the multi-pass algorithm with iterative solver.

In this research it was not possible to investigate these numbers extensively, but with a test on one model of approximately 5600 solids it was found that the engine ran 3 times faster, 11 hours instead of 33. This seems very promising because it means that within T&O that particular model can be analysed overnight instead of during a weekend. The results with single-pass are not checked on accuracy within this research.

# 5. PRACTICAL PROBLEMS

## 5.1 General

When big models are being analysed with STRUCTURE, long calculation times are to be expected. The big models within SP contain several thousand elements. An example of such a model is the drag strut of an Embraer EMB 145, a part of the nose landing gear of an aeroplane, see figure 5.1.

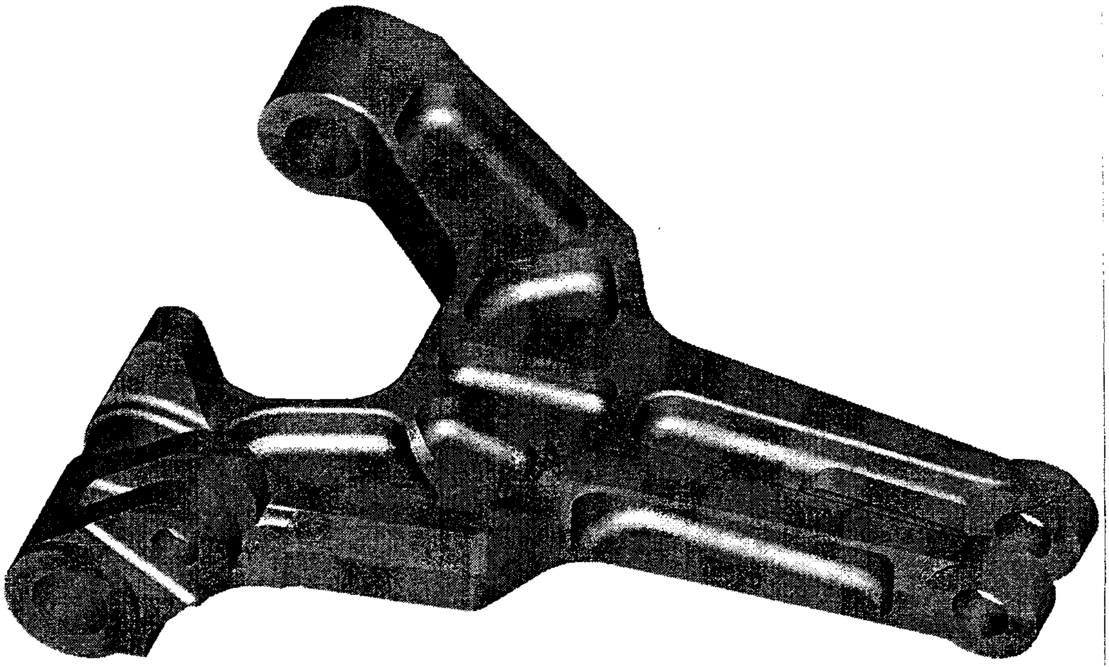


*figure 5.1: FE-model of a drag strut*

This actual model consists of approximately 5600 solid elements (tetrahedrons) and an analysis will take about 33 hours. Analysing such models during working hours is not possible since the network would be loaded to much then. The analyses have to be done overnight or during the weekend.

Although the analyses of such models are done mostly outside working hours, sometimes an analysis has to be stopped. For instance when the engineer needs results at that moment or when the network is loaded to much. When using the multi-pass algorithm, such a stop means that the FE-solution has not converged until the specified tolerance. It happened that the convergence value was still at 80% at the time of aborting the analysis. This seemed to implicate a very large deviation of the asked convergence tolerance since the convergence value needs to go from 100% towards the user specified tolerance. When actual final values were determined, using experimental techniques, it was found that the results from the aborted run were actually within 5% of the experimental results.

Such contradictions together with the long and unpredictable calculation times and the big temporary files, made it necessary to gain a better insight in the analysis methods used within STRUCTURE as described in the previous chapters. In order to try to determine the influence of some important settings in the program on the quality and accuracy of the results, as well as on calculation times and disk storage, simple models were created and analysed.

## 5.2 Banner 2-D

A model in the shape of a banner, using shells and beams, was created, see figure 5.2. The numerical values of the model and the analyses are in appendix B. The surface of the shells was loaded with a uniformly distributed load in the x-direction. The points at the bottom of the two vertical poles were fixed. For convergence several different convergence criteria were used and analysed. Also one analysis was done where the shells were excluded from upgrading after polynomial order 2. This means that the edge order of all the shells would remain at $p$=2 in spite of what the result of the convergence check is. The results for displacement and stresses for all the different settings are comparably the same and are discussed in appendix B. The size of global matrix profiles are examined, but calculation times are not examined, because the analyses are conducted on different work stations.

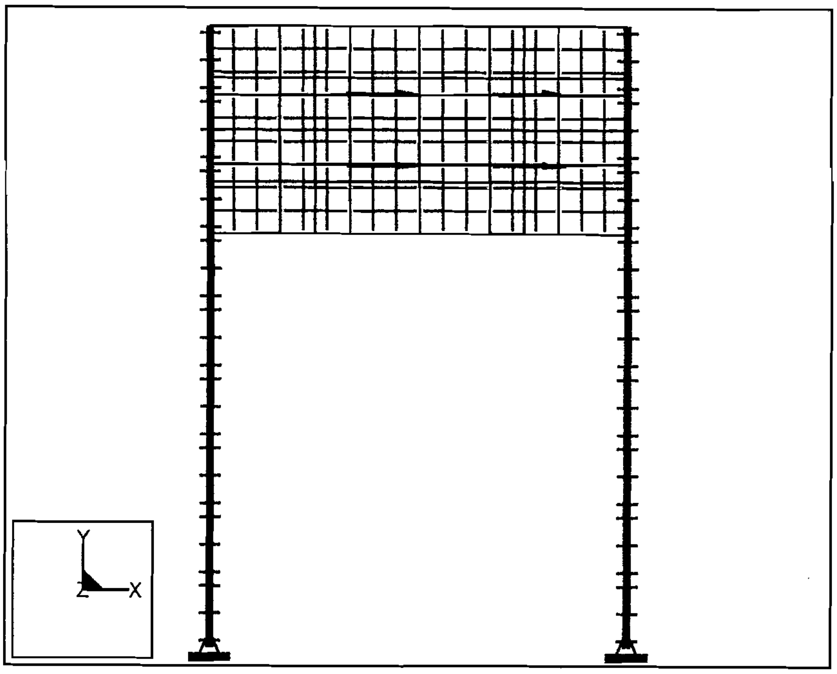


*figure 5.2: FE-model of a 2-D banner*

### *5.2.1 Local displacement & Local strain energy 1%*

When local displacement & local strain energy (LD&SE) is used for convergence, STRUCTURE calculates until convergence is reached for the two convergence criteria, for the displacements along each edge and for the strain energy of each element. With a tolerance of 1%, the analysis did not converge within the maximum polynomial order ($p$=9). The maximum edge order was reached within 2 elements. In figure B.2 in appendix B the p-levels of the different edges are visible. Here it is clearly visible that the shells are all going to edge order 4 or higher. This is not wished for since the block of shells is almost like a rigid body: the differences in displacement over the block are very small and the stresses within this block are also small. At the connection of the vertical beams and the shells there is a high gradient in strain energy which makes it understandable that a higher polynomial order is needed in those elements in comparison to the elements in the middle of the banner. The higher order elements in the middle of the banner are a result of the big displacements for those elements. Since the upgrading of the elements is based on displacements and strain energies, as discussed in chapter 4, the edge orders in the middle will go up due to the large displacements. These higher orders for the shells occur in practice, but according to reference [1], there should not have been such an increase for the polynomial orders of the shells, since the displacement of the shells can be seen as a rigid body motion. As described in paragraph 4.2.2 under point 8.a), unnecessary updates due to rigid body motion should be accounted for. Where this discrepancy originates from, was not found during this study. The size of the global matrix profile for the final pass was 862 kb.

A way of keeping the edge orders lower in the middle of the banner was looked for. It was tried with two other convergence criteria: a measure on x-displacement and a measure on Von Mises stress.

### *5.2.2 Measure 1: 1% x-displacement*

A measure was placed on one of the vertical poles, a measure for the displacement in the x-direction with a tolerance of 1%. The p-levels for this analysis are shown in figure B.3. This time the analysis did convergence. Polynomial order 5 was needed for the shells where the poles and the banner are connected. Again the high polynomial orders for the rest of the shells was found (p = 4), even though the measure was placed on a vertical pole. This is possible since the convergence check and the upgrading are two separate things. The convergence check is committed on the x-displacement for a point on a vertical pole, but when convergence is not yet achieved, the upgrading of the element edges is done with local displacement & local strain energy, as described in chapter 4. The large displacements in the banner still give the high polynomial orders in that banner. The size of the global matrix profile for the final pass was 413 kb. This is evidently much smaller than with the LD&SE criteria. The reason for this is the lower *p*-order reached after convergence. With LD&SE order 9 was achieved and here only order 5 was reached. Lesser shape functions and coefficients had to be used and calculated which also means that the matrices stay smaller in comparison to the LD&SE criteria.

### *5.2.3 Measure 2: 1% Von Mises stress*

At one of the fixed points, a measure for the Von Mises stress was placed with a tolerance of 1%. The same distribution of the p-levels over the different element edges as with measure 1 occurred when an analysis was done. Here again the upgrade scheme as discussed in chapter 4 is responsible for the high p-levels within the banner. The size of the global matrix profile for the final pass was 413 kb, exactly the same as with measure 1.

### *5.2.4 Excluded elements*

This analysis was done using the LD&SE convergence criteria set to 1%. The shells were excluded from convergence after polynomial order 2. Obviously the orders in the banner are low for this analysis, as we tried to accomplish with the previous convergence criteria. Excluding of elements, however, is dissuaded by the supplier of STRUCTURE. This is understandable because with excluding the elements from upgrading, you influence the model. In this particular example the results are comparably the same as when you don't exclude, see appendix B. The size of the global matrix profile for the final pass was only 58 kb. The small size of the matrix for this analysis is understandable since only the beams reach high orders, until order 4. The shells are excluded from convergence after order 2. Therefore there are no new shape functions needed for the shells and the matrices stay relatively small.

## 5.3 Banner 3-D

Within T&O of Daf SP most of the models are imported from the CAD-program Pro/ENGINEER. The models are build with volumes and when the automatic mesher (AutoGEM) is used, solid elements (tetrahedrons) are created. This is the reason why a three dimensional model was created and meshed using AutoGEM, see figure 5.3. For this model again a banner was used. The two surfaces of the banner in the xy-view are loaded with an uniformly distributed load in the x-direction. The bottom surfaces of the two vertical poles are fixed. In appendix C the numerical values of the model and the results are shown. With the 3-D banner the results in displacement and stresses are comparably the same for all three analysis and shown in appendix C. Again the calculation times could not be checked because the analysis are conducted on different work stations.

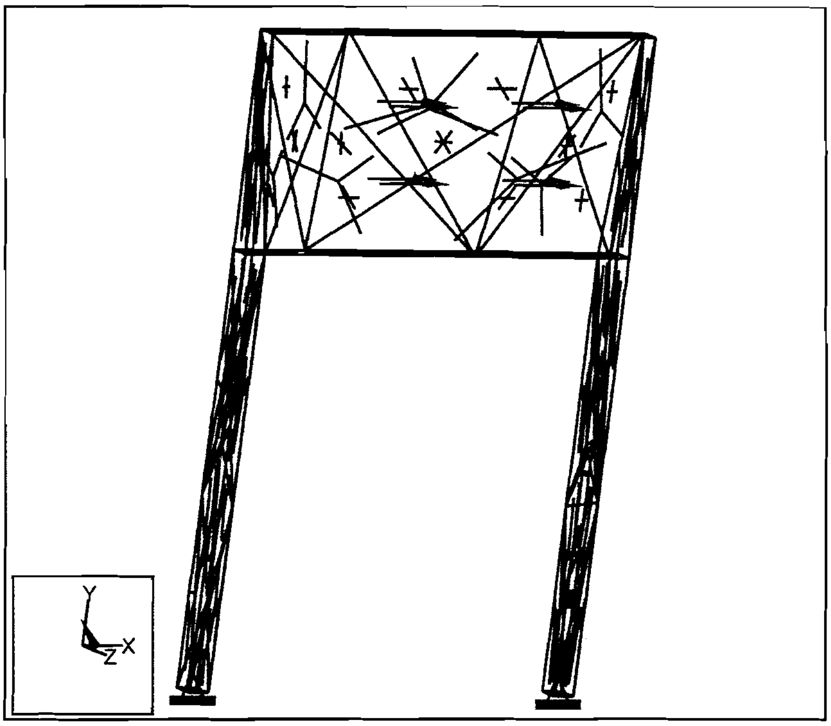


*figure 5.3: FE-model of a 3-D banner*

The model was analysed using both the convergence algorithm's: multi-pass as well as single-pass. With multi-pass different convergence criteria were used: LD&SE at 5% and LD&SE&RMS at 5%.

### *5.3.1 LD & SE 5%*

For convergence the multi-pass algorithm with the criteria LD&SE was used, with a tolerance of 5%. The run was completed successfully and the p-level plot after convergence is shown in appendix C, figure C.3. In this plot it is clearly visible that the edge orders in the banner are small in comparison to the edge orders in the vertical poles. This is what was expected since the high deformations are in the vertical poles and not in the banner. As discussed with the 2-D banner, the banner can be considered a rigid body in comparison to the vertical poles. At the connection of the poles and the banner the edge orders are higher as in the rest of the model. This was expected as well since at those points the largest gradients in the stresses and strains are available. The size of the global matrix profile for the final pass was 6.462 Mb.

### 5.3.2 LD & SE & RMS 5%

Again the multi-pass convergence algorithm was used. For a convergence criteria, LD&SE&RMS was used. and the run was completed successfully. As mentioned earlier, the RMS stress over the entire model is now taken into account in addition to local displacement and local strain energy. The p-level plot in appendix C, figure C.5, shows a similar distribution of the edge orders as with LD&SE, only for higher levels. Again the vertical poles show higher p-levels than the banner and the highest edge orders are at the connection of the poles with the banner. The size of the global matrix profile for the final pass was 33.611 Mb. This is much bigger than the global matrix profile when LD&SE is used as a convergence criteria. The big size is originating from the higher edge orders which occur. For instance the edge orders in the vertical poles. In the vertical poles for the LD&SE criteria, most of the edge orders are $p$=4, and for the LD&SE&RMS criteria, the edge orders are $p$=8. This is clearly visible in appendix C, the figures C.3 and C.5.

### 5.3.3 Single-pass

When the single-pass convergence algorithm is used to analyse the 3-D banner, the p-level distribution over the model is again of the same form as with the multi-pass algorithm, see appendix C, figure C.7. The p-levels are lower as with multi-pass. Within the vertical poles the edge orders only are upgraded there were large stress/strain gradients are present. In the middle of the poles, where the deformations are small, the edge orders are staying at order three. The size of the global matrix profile for the final pass was 11.284 Mb. This is hard to compare with the two multi-pass analyses. When the matrix sizes are looked at as an indication of the calculation times, then it should be taken in account that the set of equations with single-pass does not need to be solved so many times as with multi-pass. Even when the matrices are bigger then, the calculation times can still be smaller.

## 5.4 Conclusions for practical problems

When a model is analysed with a part of it which can be considered as a rigid body, shells seem to be upgraded more than solids. This upgrading is due to the upgrading scheme that is used within STRUCTURE, described in chapter 4. The large displacement of the shells delivers the higher p-levels in those shells. The solid elements do not show such a property, although with the examined 3-D model there was a rigid body displacement as well. It seems that the upgrading scheme for solids is different from that of the shells. This also was mentioned as an idea by the supplier of STRUCTURE. Such a different upgrading scheme was not found in the manuals or in literature.

The single-pass algorithm also keeps lower p-levels, in the part of the examined 3-D model which can be looked at as a rigid body. Between the multi-pass and the single-pass analyses of the 3-D model there could not be found a significant difference in the distribution of the p-levels.

# 6. CONCLUSIONS & RECOMMENDATIONS

## 6.1 Conclusions

As it is claimed by the developer of STRUCTURE, the program is very user-friendly.

The theory of the p-method, as it is implemented in STRUCTURE, is well described within this report. It is shown that not all the advantages of the p-method, with hierarchical shape functions based on the integrals of Legendre polynomials, are used. For instance, the expected diagonality of the stiffness matrix does not occur with large models in practice.

Finding the right models to test settings of the program, is found to be a very difficult subject. But with the used models a few properties are found:

- Solid elements seem to have a better behaviour in comparison to shells when a 'rigid body' is analysed.
- Settings for lowering the calculation times and the disk storage capacity for the multi-pass algorithm are not found.
- The single-pass algorithm seems to give a large reduction in calculation times, in comparison to the multi-pass algorithm.

## 6.2 Recommendations

In order to get a better understanding of the long calculation times, it should be investigated why the matrix diagonality does not occur.

Investigate the accuracy of the results of the single-pass algorithm, in order to make more use of this convergence algorithm. To get a better understanding of this algorithm, the stress error estimation theory used, the $Z^2$ method of Zienkiewicz and Zhu, should be examined more theoretically.

To decrease the calculation time of an analysis, it is recommended to perform a calculation locally. This means that the temporary files as well as the result files should be written to the local hard disk. This to avoid the data traffic over the network, which is considerably slower than the data traffic internally in a work station.

The large temporary files in the order of Gigabytes, produced during an analysis, are a permanent property of STRUCTURE when big models are calculated., and seems unavoidable.

# 7. LITERATURE

1. Applied Structure Theoretical Overview, Applied Structure Version 2.1. Rasna Corporation (May 1992), San Jose.
2. Babuška, I., Szabó, B.A., and Katz, I.N. (1981). "The p-Version of the Finite Element Method," SIAM Journal of Numerical Analysis, vol. 18, 515-545.
3. Babuška, I., and Szabó, B.A. (1982). "On the Rates of Convergence of the Finite Element Method," International Journal of Numerical Methods, vol. 18, 323-341.
4. Szabó, B., Babuška, I., (1991). Finite Element Analysis, John Wiley & Sons, New York
5. Zienkiewicz, O.C., Taylor, R.C., (1989). The Finite Element Method, vol 1, McGraw-Hill book company, London.
6. Zienkiewicz, O.C., Zhu, J.Z., (1992). "The Superconvergent Patch Recovery and A Posteriori Error Estimates. Part 1: The Recovery Technique", International Journal for Numerical Methods in Engineering, vol. 33, p. 1331.
7. Young, Warren C., (1989). Roark's formulas for stress & strain, sixth edition, McGraw-Hill, Inc.
8. Pro/MECHANICA STRUCTURE™ User's Guide for release 8.0. Parametric Technology Corporation (September 1995).

# 8. APPENDIX A: LEGENDRE POLYNOMIALS

This description of the Legendre polynomials is taken from reference [4]. The Legendre polynomials $P_n(x)$ are solutions of the Legendre differential equation for $n = 0,1,2,\ldots$ :

$$\left(1-x^2\right)y''-2xy'+n(n+1)y=0, \qquad -1 \le x \le 1 \tag{A.1}$$

The first eight Legendre polynomials are:

$$P_0(x)=1 \tag{A.2.a}$$

$$P_1(x)=x \tag{A.2.b}$$

$$P_2(x)=\frac{1}{2}\left(3x^2-1\right) \tag{A.2.c}$$

$$P_3(x)=\frac{1}{2}\left(5x^3-3x\right) \tag{A.2.d}$$

$$P_4(x)=\frac{1}{8}\left(35x^4-30x^2+3\right) \tag{A.2.e}$$

$$P_5(x)=\frac{1}{8}\left(63x^5-70x^3+15x\right) \tag{A.2.f}$$

$$P_6(x)=\frac{1}{16}\left(231x^6-315x^4+105x^2-5\right) \tag{A.2.g}$$

$$P_7(x)=\frac{1}{16}\left(429x^7-693x^5+315x^3-35x\right) \tag{A.2.h}$$

Legendre polynomials can be generated from Bonnet's recursion formula:

$$(n+1)p_{n+1}(x)=(2n+1)xP_n(x)-nP_{n-1}(x), \qquad n=1,2,\ldots \tag{A.3}$$

and Legendre polynomials satisfy the following relationship:

$$(2n+1)p_n(x)=P'_{n+1}(x)-P'_{n-1}(x), \qquad n=1,2,\ldots \tag{A.4}$$

where the primes represent differentiation with respect to $x$. Legendre polynomials satisfy the following orthogonality property:

$$\int_{-1}^{1} P_i(x)P_j(x)dx=\begin{cases}\dfrac{2}{2i+1} & \text{for } i=j \\ 0 & \text{for } i\neq j\end{cases} \tag{A.5}$$

All roots of Legendre polynomials occur in the interval $-1 < x < +1$. The $n$ roots of $P_n(x)$ are the abscissas $x_i$ for the $n$-point Gaussian integration:

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{A.6}$$

The abscissas $x_i$ and weight factors $w_i$ for Gaussian integration are listed in table A.1.

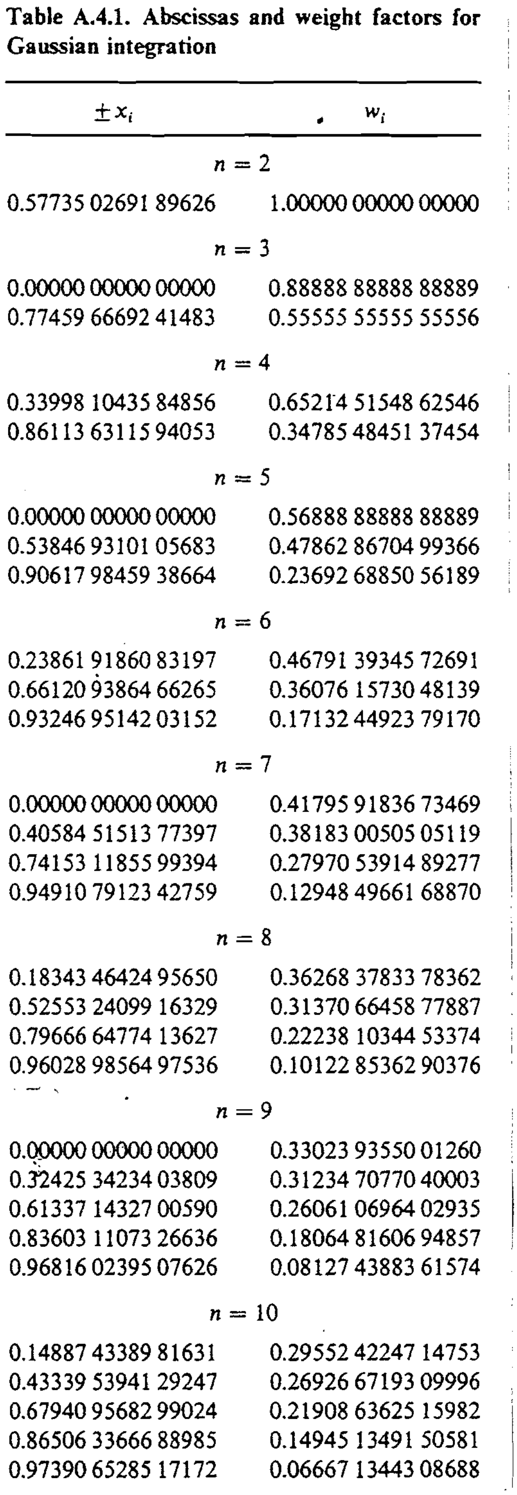


**Table A.4.1. Abscissas and weight factors for Gaussian integration**

| $\pm x_i$ | $w_i$ |
|---|---|
| $n = 2$ | |
| 0.57735 02691 89626 | 1.00000 00000 00000 |
| $n = 3$ | |
| 0.00000 00000 00000 | 0.88888 88888 88889 |
| 0.77459 66692 41483 | 0.55555 55555 55556 |
| $n = 4$ | |
| 0.33998 10435 84856 | 0.65214 51548 62546 |
| 0.86113 63115 94053 | 0.34785 48451 37454 |
| $n = 5$ | |
| 0.00000 00000 00000 | 0.56888 88888 88889 |
| 0.53846 93101 05683 | 0.47862 86704 99366 |
| 0.90617 98459 38664 | 0.23692 68850 56189 |
| $n = 6$ | |
| 0.23861 91860 83197 | 0.46791 39345 72691 |
| 0.66120 93864 66265 | 0.36076 15730 48139 |
| 0.93246 95142 03152 | 0.17132 44923 79170 |
| $n = 7$ | |
| 0.00000 00000 00000 | 0.41795 91836 73469 |
| 0.40584 51513 77397 | 0.38183 00505 05119 |
| 0.74153 11855 99394 | 0.27970 53914 89277 |
| 0.94910 79123 42759 | 0.12948 49661 68870 |
| $n = 8$ | |
| 0.18343 46424 95650 | 0.36268 37833 78362 |
| 0.52553 24099 16329 | 0.31370 66458 77887 |
| 0.79666 64774 13627 | 0.22238 10344 53374 |
| 0.96028 98564 97536 | 0.10122 85362 90376 |
| $n = 9$ | |
| 0.00000 00000 00000 | 0.33023 93550 01260 |
| 0.32425 34234 03809 | 0.31234 70770 40003 |
| 0.61337 14327 00590 | 0.26061 06964 02935 |
| 0.83603 11073 26636 | 0.18064 81606 94857 |
| 0.96816 02395 07626 | 0.08127 43883 61574 |
| $n = 10$ | |
| 0.14887 43389 81631 | 0.29552 42247 14753 |
| 0.43339 53941 29247 | 0.26926 67193 09996 |
| 0.67940 95682 99024 | 0.21908 63625 15982 |
| 0.86506 33666 88985 | 0.14945 13491 50581 |
| 0.97390 65285 17172 | 0.06667 13443 08688 |

*table A.1: Abscissas and weight factors for Gaussian integration*

# 9. APPENDIX B: BANNER 2-D

The model of the 2-D banner is shown in figure B.1 and in figure 5.2 a FE-model.

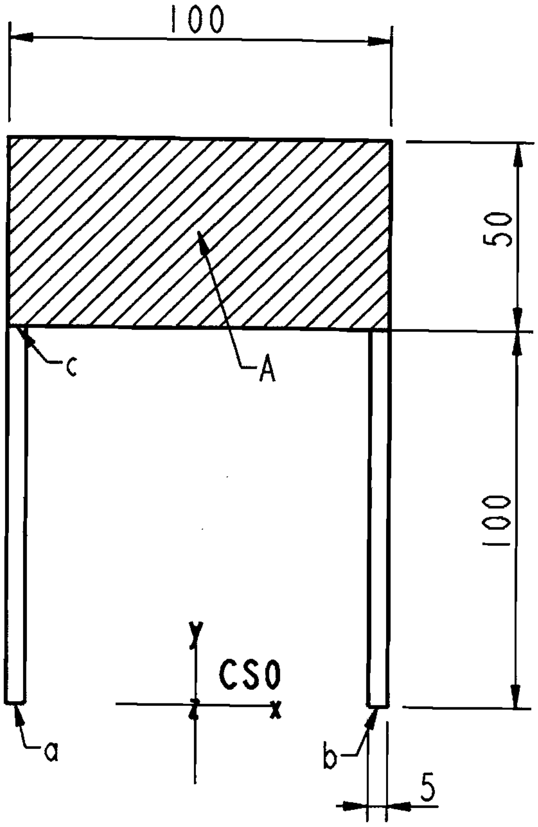


*figure B.1: Model of a 2-D banner*

## Properties:

*Constraints*:

The points a and b are constrained in all degrees of freedom, these points are fixed to the ground.

*Loads*:

The surface A is loaded with an uniformly distributed load in the x-direction, FX = 100.

*Material*:

The material is steel with a young's modulus of 199900 and a poisson's ratio of 0,27.

## Analytical results:

When the surface a is looked as a rigid body, the deflection in the x-direction of the point b on the one of the vertical poles can be determined using the Roark's formula [7] for a beam which is simply supported at the one end and fixed at the other end. This formula is:

$$x = \frac{F \cdot l^3}{12 \cdot E \cdot I} \qquad \text{(B.1)}$$

in which:

F: Half of the load due to symmetry, F = 50

l: Length of the pole, l = 100

E: Young's modulus, E = 199900

I: Area moment of inertia about the centroidal axis of the beam section, $I = \frac{a^4}{12}$

a: Length of the sides of the pole, a = 5

This leads to a deflection of: x = 0,4002

## Mesh of elements:

*Beams*:

Alongside of the vertical poles and the short side of the surface, eighteen beams were placed. Six beams on each vertical pole and three along each of the short sides of the surface.

*Shells*:

On surface A eighteen rectangular shells were placed. The long side of the surface was divided in six sections and the short side in three. The properties of the shells and the beams are shown in table B.1.

| | |
|---|---|
| beams | square, length of sides = 5 |
| shells | thickness = 5 |

*table B.1: Properties of the elements*

## Analyses:

*Local displacement & local strain energy, 1%*

In figure B.2 fringe plots for the p-levels, the Von Mises stress, and the displacements are visualised.

*Measure on x-displacement, 1%*

In figure B.3 fringe plots for the p-levels, the Von Mises stress, and the displacements are visualised.

*Excluded elements*

In figure B.4 fringe plots for the p-levels, the Von Mises stress, and the displacements are visualised.

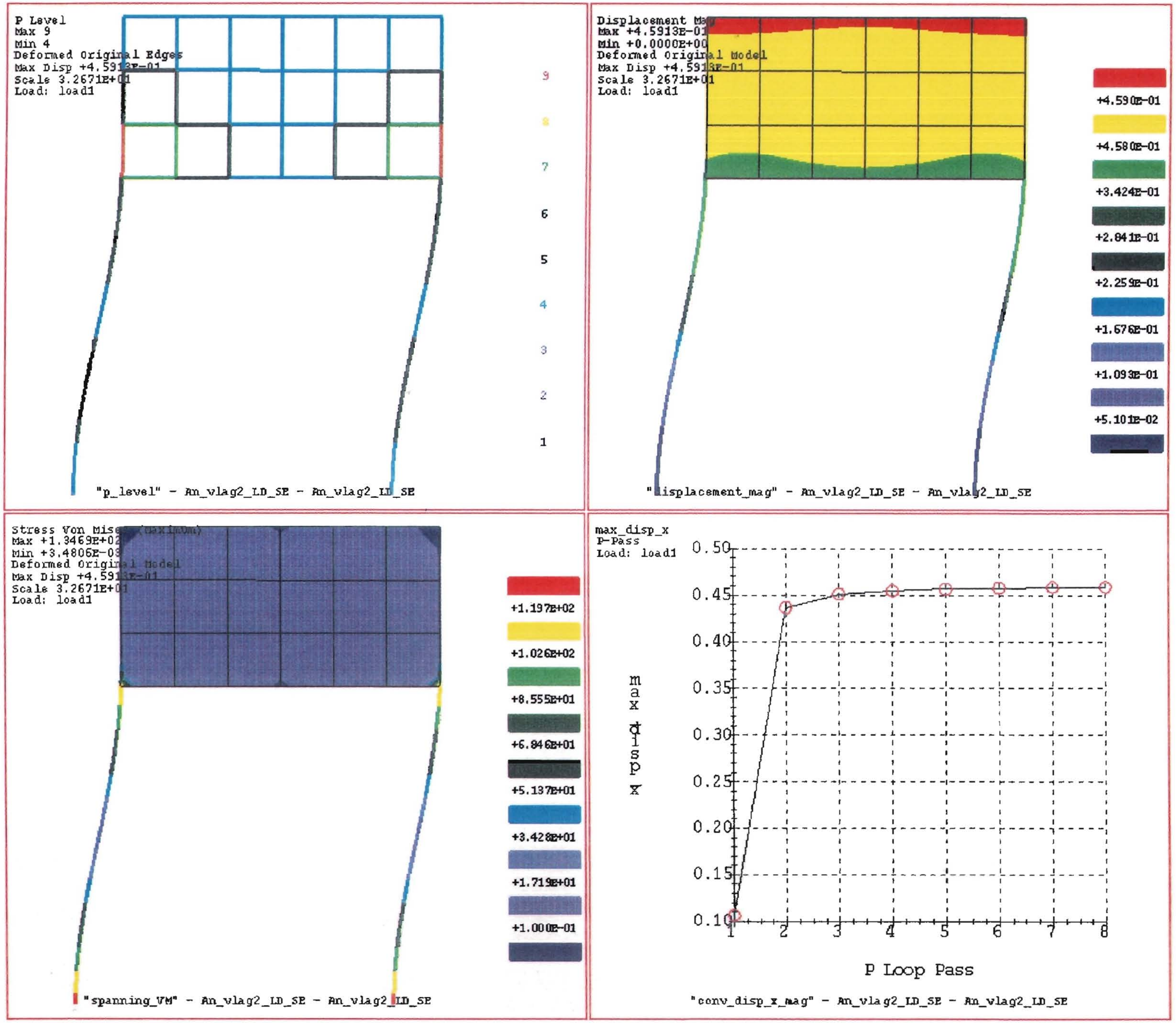


*figure B.2: Results of the LD&SE 1% analysis on the 2-D banner*

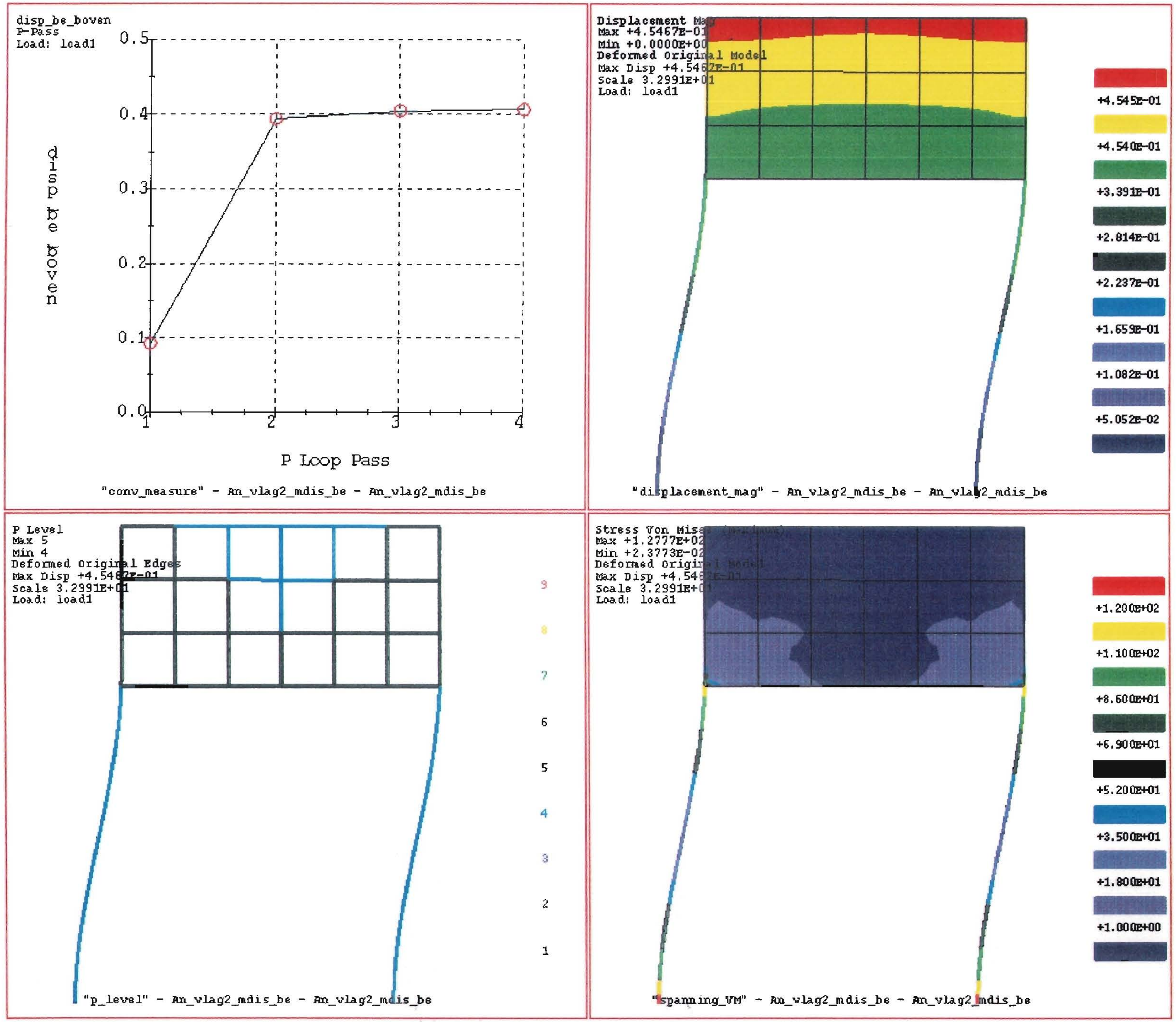


*figure B.3: Results of the x-displacement 1% analysis on the 2-D banner*

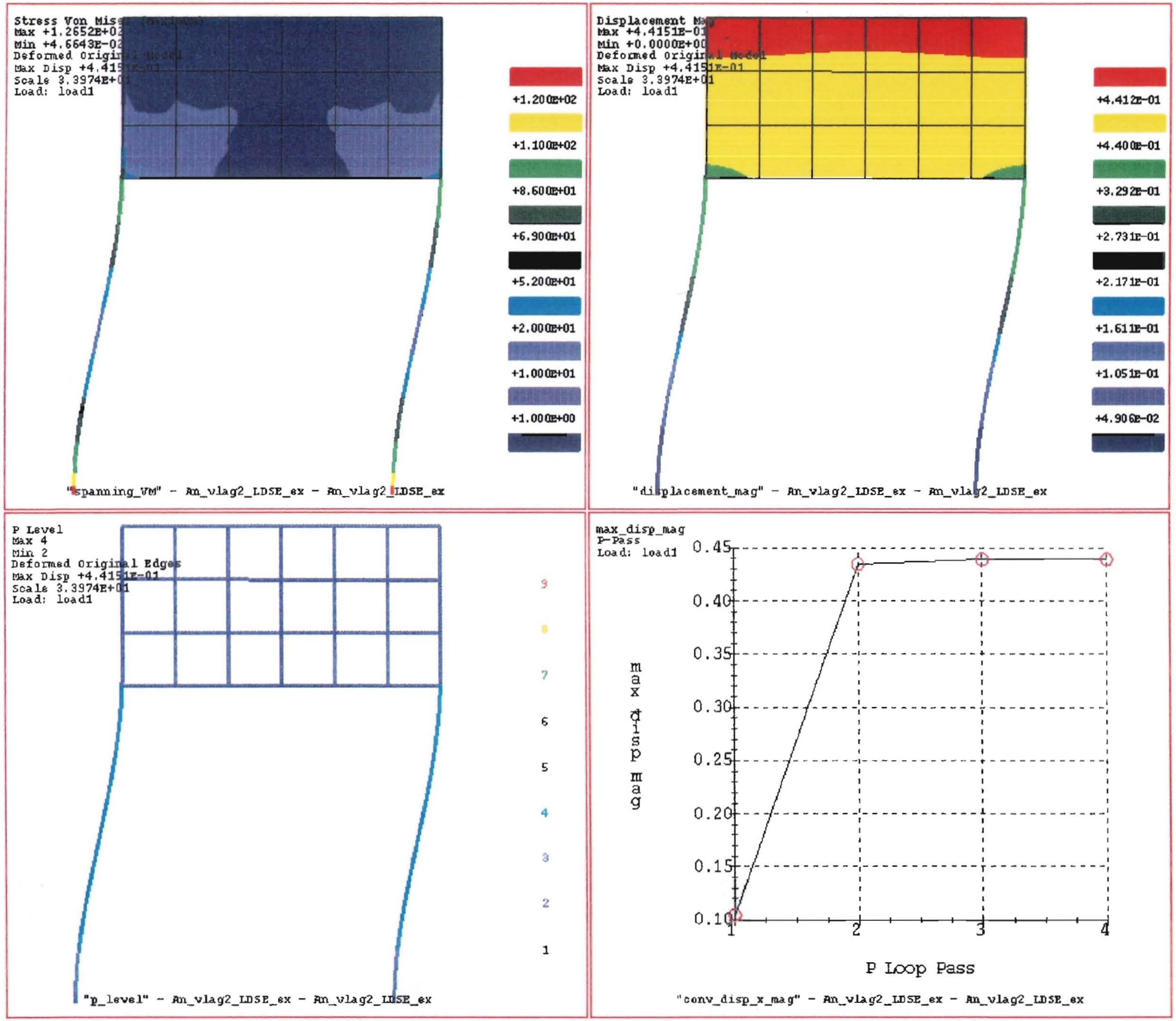


figure B.4: Results of the LD&SE 1% analysis, with excluded elements, on the 2-D banner

# 10. APPENDIX C: BANNER 3-D

The model of the 3-D banner is shown in figure C.1 and in figure 5.3 a FE-model.

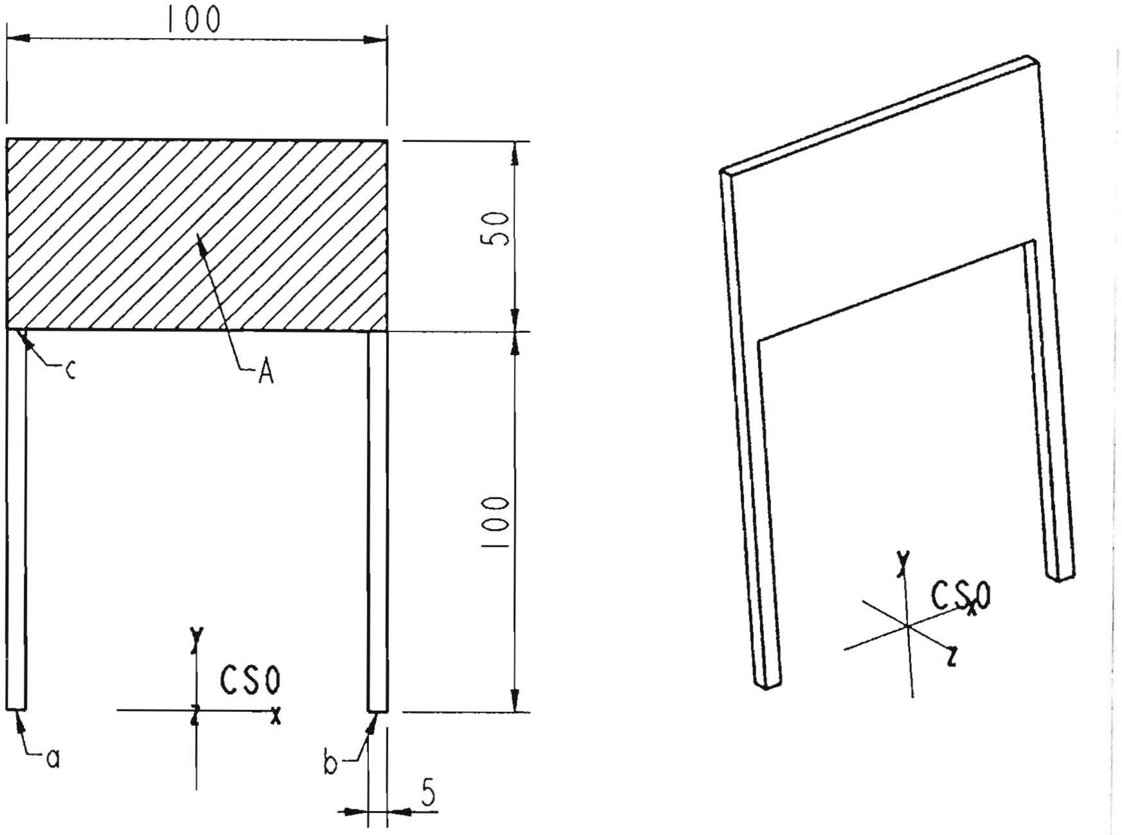


*figure C.1: Model of a 3-D banner*

The thickness in the z-direction is 5.

## Properties:

*Constraints*:

The ground surfaces a and b are constrained in all degrees of freedom, these surfaces are fixed to the ground.

*Loads*:

Both the front and back surfaces of the banner are loaded with an uniformly distributed load in the x-direction. They both are FX = 50.

*Material*:

The material is steel with a young's modulus of 199900 and a poisson's ratio of 0,27.

## Mesh of elements:

For generating a mesh of elements, the automatic mesher within STRUCTURE is used. Using this feature, 44 solid elements were created, all tetrahedrons.

## Analyses:

### *Local displacement & local strain energy, 5%*

In figure C.2 fringe plots for the p-levels, the Von Mises stress, and the displacements are visualised. Figure C.3 shows a larger plot of the p-levels so every edge is visible.

### *Measure on x-displacement, 1%*

In figure C.4 fringe plots for the p-levels, the Von Mises stress, and the displacements are visualised. Figure C.5 shows a larger plot of the p-levels so every edge is visible.

### *Excluded elements*

In figure C.6 fringe plots for the p-levels, the Von Mises stress, and the displacements are visualised. Figure C.7 shows a larger plot of the p-levels so every edge is visible.

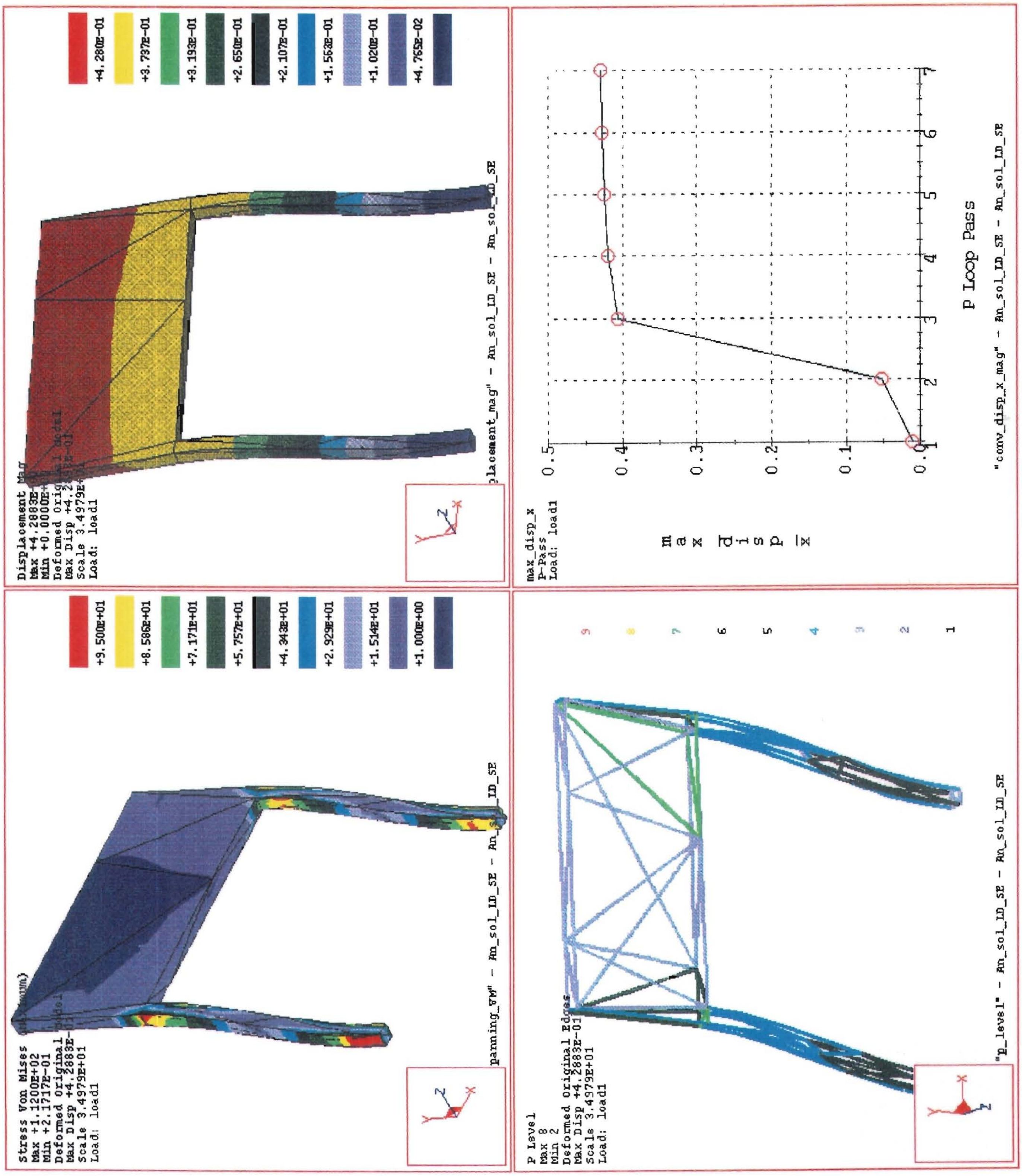


*figure C.2: Results of the LD&SE 5% analysis on the 3-D banner*

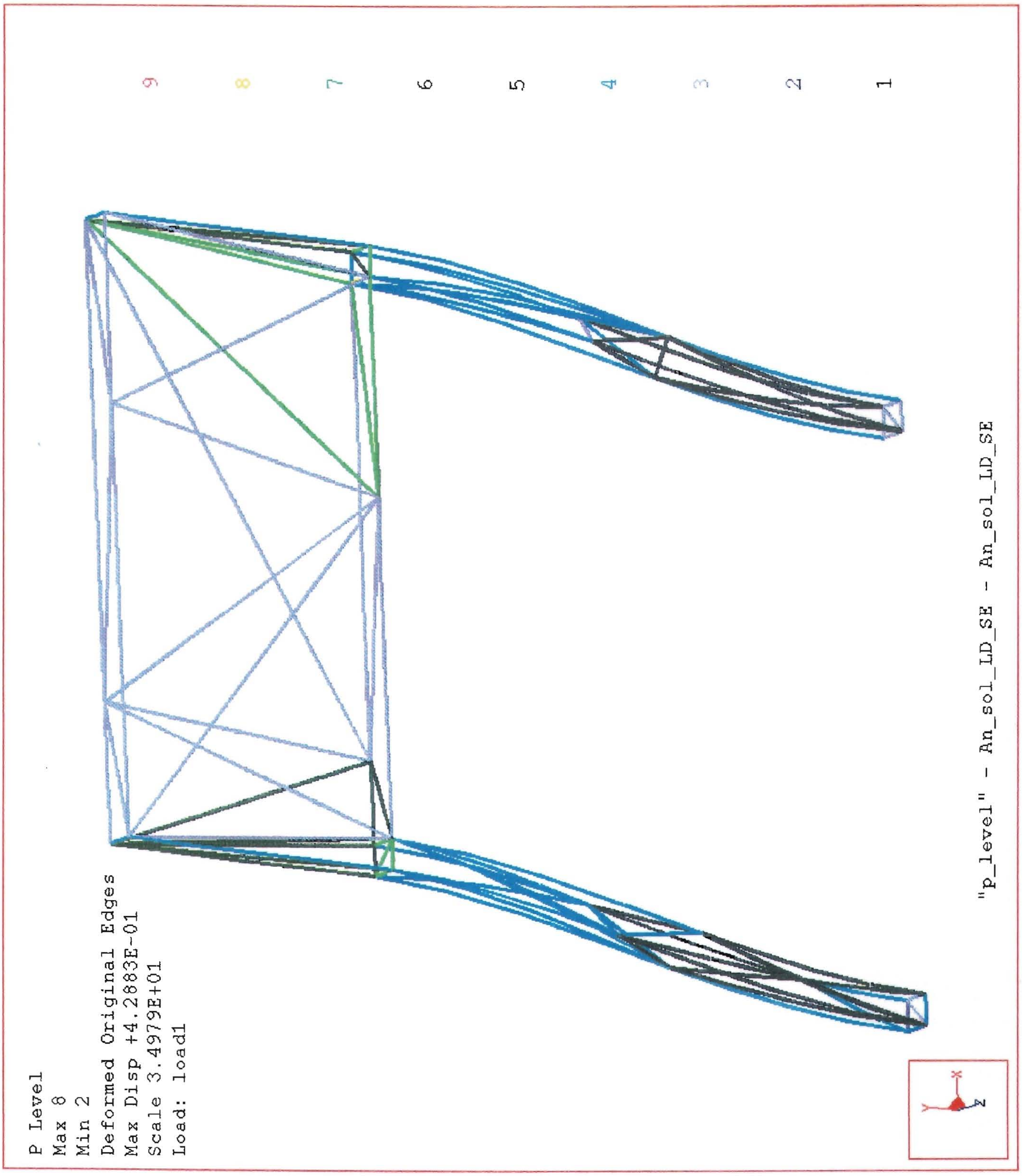


*figure C.3: P-level plot of the LD&SE 5% analysis on the 3-D banner*

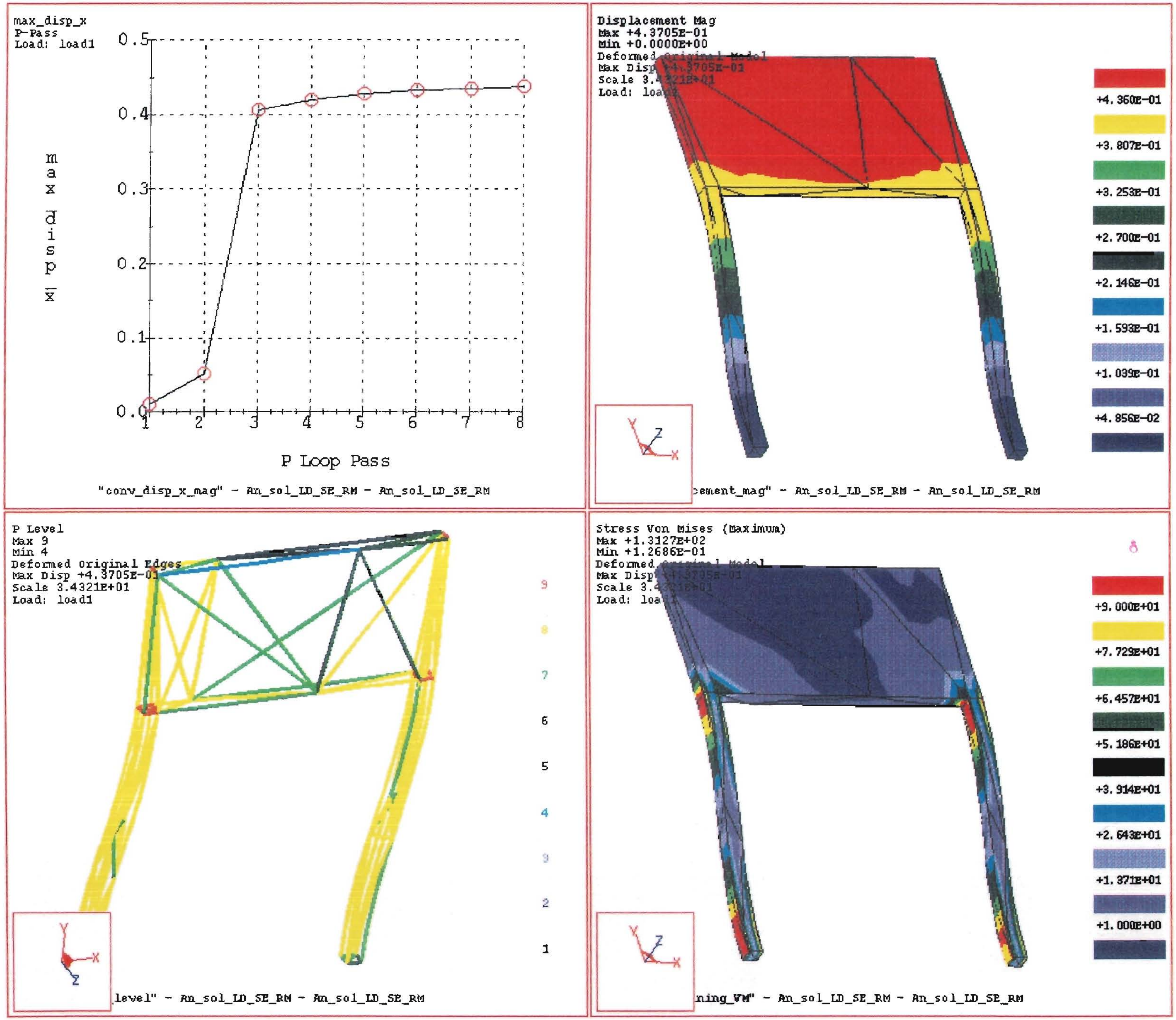


*figure C.4: Results of the LD&SE&RMS 5% analysis on the 3-D banner*

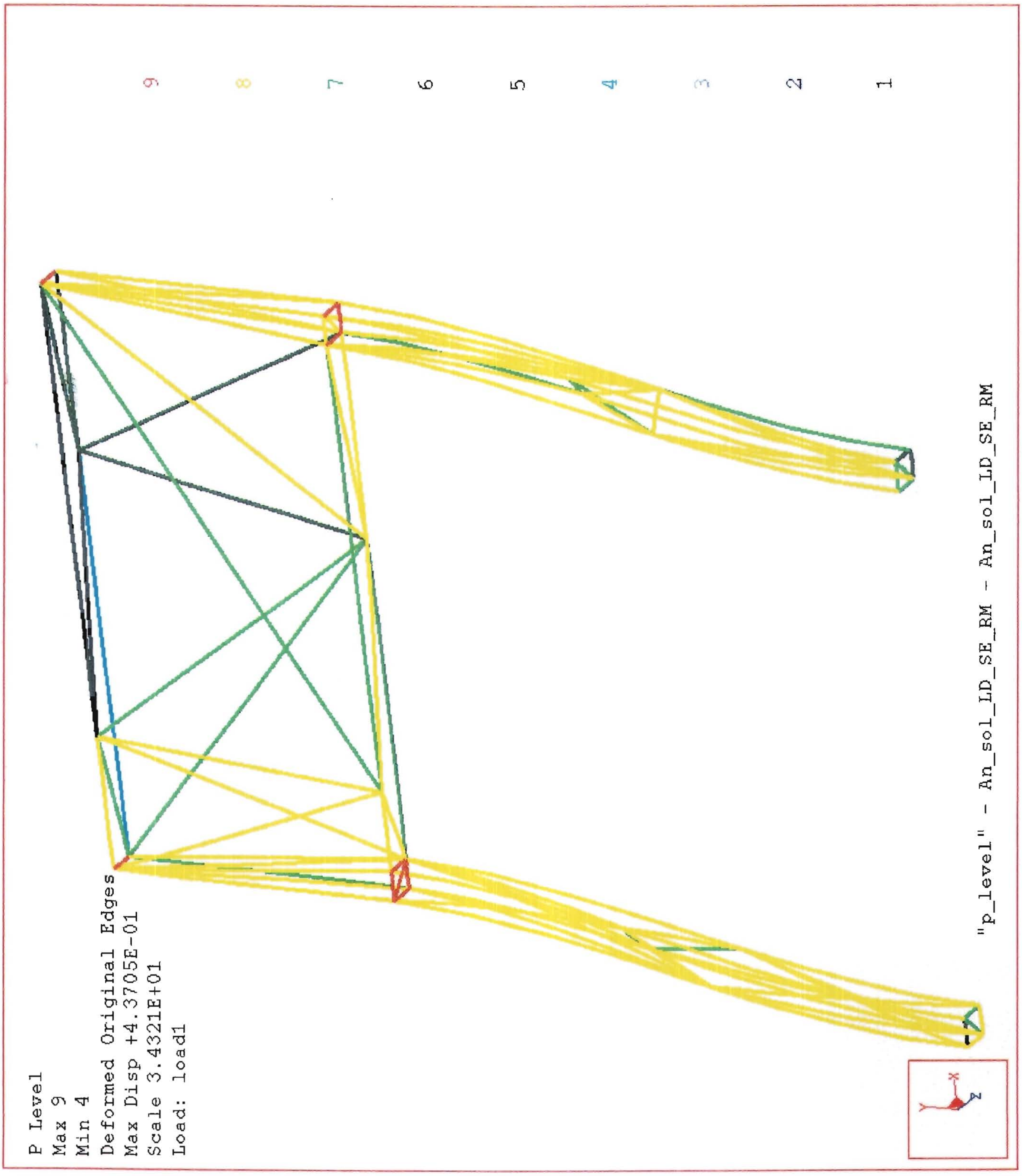


*figure C.5: P-level plot of the LD&SE&RMS 5% analysis on the 3-D banner*

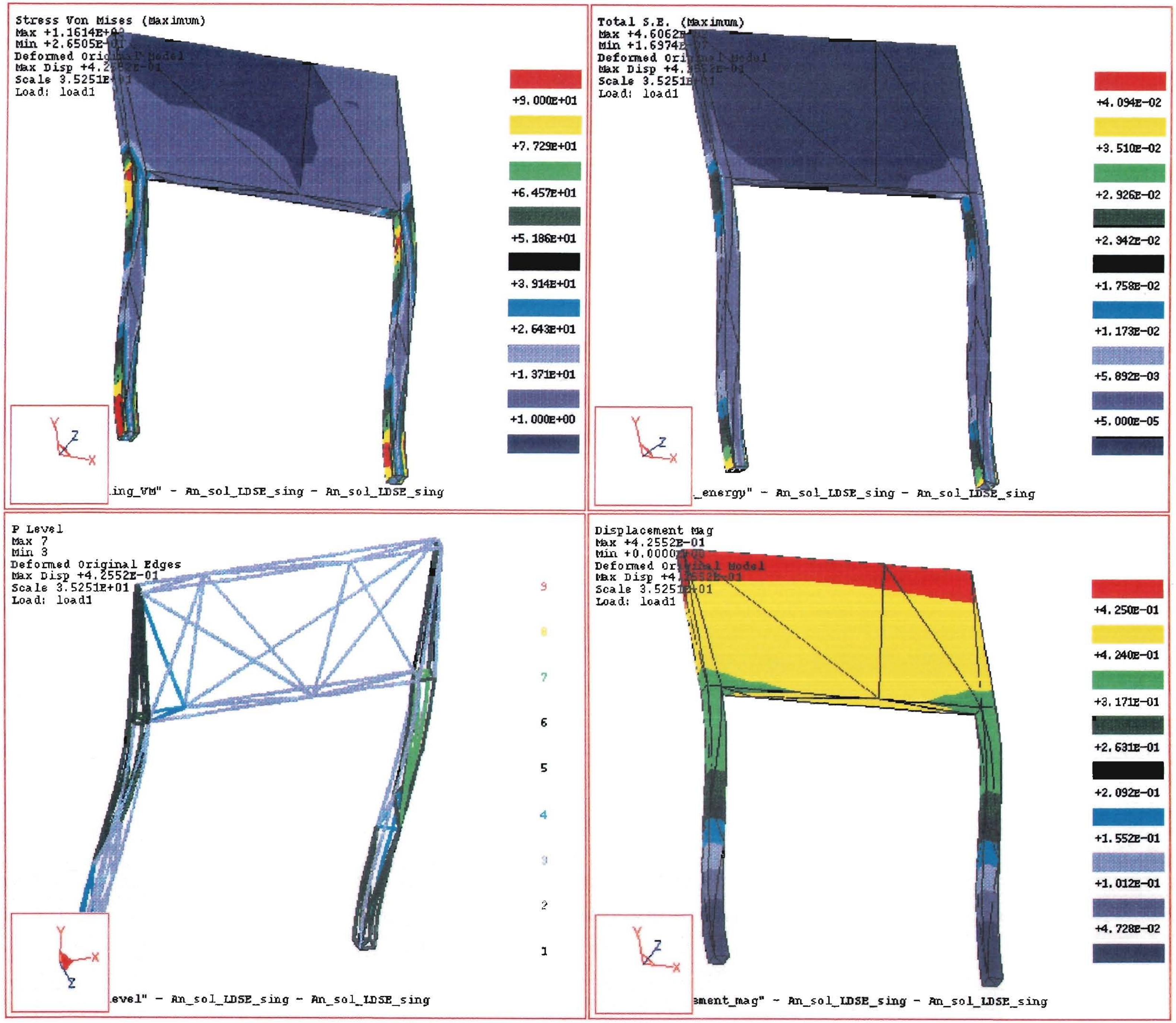


*figure C.6: Results of single-pass analysis on the 3-D banner*

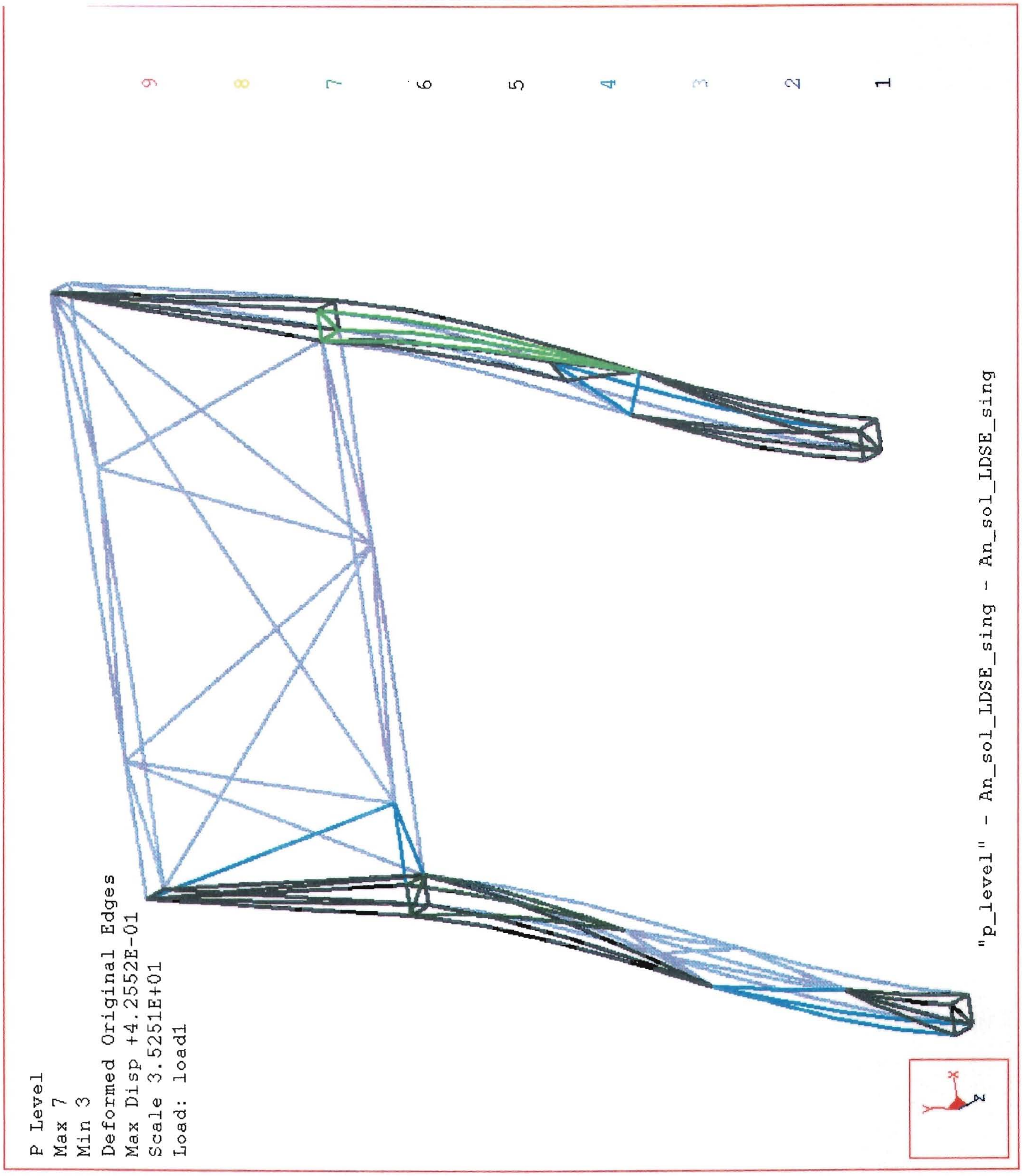


*figure C.7: P-level plot of the single-pass analysis on the 3-D banner*