Computer Assisted Mechanics and Engineering Sciences, 5: 101–110, 1998.
Copyright © 1998 by Polska Akademia Nauk

Advanced finite elements in civil engineering practice

Stefan M. Holzer
Informationsverarbeitung im Konstruktiven Ingenieurbau, Universität Stuttgart, Pfaffenwaldring 7, D-70550 Stuttgart, Germany

(Received December 20, 1996)

New developments in structural analysis methods such as automatic error estimation, adaptive discretization control, and mesh generation are slowly becoming available in engineering practice. On the other hand, graphical interactive user environments and easy-to-use general purpose programs are prolific. Each of these new developments helps to ease the everyday work of the engineer, but only by joining them together in a unified framework and user environment, an entirely new class and generation of structural analysis tools on the computer can be generated. The present study suggests guidelines and principles how such a new generation of software can be brought about, and reports experiences with a prototype application that is already in practical use.

1. Computer integrated structural analysis in civil engineering

By tradition, structural analysts in civil engineering use computers as a tool for numerical computations and "number crunching". However, computers have invaded today's offices, and more and more everyday tasks, beyond the pure numerical computations, are being achieved by the aid of the desktop computer, including many tasks that used to be carried out only by paper and pencil only a few years ago. Furthermore, the technological development in peripheral devices such as printers and screens has led to customers demanding increased quality standards; therefore, graphical presentations and also ordinary texts need to be prepared on a computer system in order to fulfil the customers' expectations.

The pervasiveness of computers in the world of labour has brought up new problems, especially the problem of integrating and fitting together all those diverse and independent computer-based tools that have become available in the recent years. Up till now, the idea of improved data exchange has been considered most prominent and urging, following the idea that cost and errors introduced by redundant input of the same data could easily be avoided by improved interconnections between individual programs.

However, improved data flow between different software products is but one of many aspects of integration, improved efficiency, and quality assurance in the computer aided office.

Another, equally important, aspect is creating a unified user interface for diffent programs. The average user switches frequently between various applications and does not focus on a single program most of the time. Errors and inefficient, slow work progress will result if those applications do not have similar front ends and functionality ("look and feel").

Apart from a unified graphical appearance and easy-to-learn functionality, engineering software tools should offer user interfaces designed in accordance with engineering terms and traditional engineering models; manual computational and plausibility checks ought to be facilitated.

The present paper outlines strategies to exploit modern advanced numerical methods and computer science in order to create programs for structural analysis that match the needs of the practitioners better than today's state-of-the-art software.

2. Designing advanced structural analysis software

Structural analysis is normally performed in three steps, starting with an abstraction of the real-world structure to a structural model. Next, the structural model will be subject to a numerical analysis, and finally, relevant results need to be filtered out and taken as a basis for the dimensioning and design.

Each of these three steps makes use of a specific model of the original structure

1. structural models,
2. numerical models,
3. result evaluation models.

These three methods seem to be similar at the first glance, each of them being based on a geometrical representation of the structure; however, their purpose is entirely different. The structural model (1) contains essential assumptions about the mechanics and functionality of the structure, whereas the numerical models (2) are only an auxiliary discretization necessary to obtain numerical data. However, the numerical models are strictly depending on the structural model and cannot be chosen indepently. Similarly, the result evaluation models are also auxiliary models, e.g. meshes that facilitate the graphical evaluation of the results, including the location of stress maxima and design-critical areas of the structure. In the recent past, those three model types have not been well distinguished, and many commercial programs that are being used today in practice are founded exclusively on the numerical model, whereas all data referring to the structural and result models are being viewed as simple attributes to the numerical model.

In each of the modeling steps indicated, new simplifications and errors are introduced. These errors need to be controlled in order to ensure the overall quality of the analysis. None of the three model categories contains only one single, "correct" model, but a suitable model must be selected from a wide range of possibilities. This selection needs to be guided by engineering judgement as well as automatic, a priori and a posteriori, knowledge extraction.

2.1. Automatical generation of the structural model

In the first modeling step, the analyst creates a mathematical abstraction of the physical reality (geometry, material behaviour, loads). The corresponding abstraction of the real structure is laid down in a "structural model". The structural model is based on a simplified geometrical and topological representation of the real structure. To some extent, construction of this model may be supported by data imported from other geometrical models of the structure, e.g., from a CAD model.

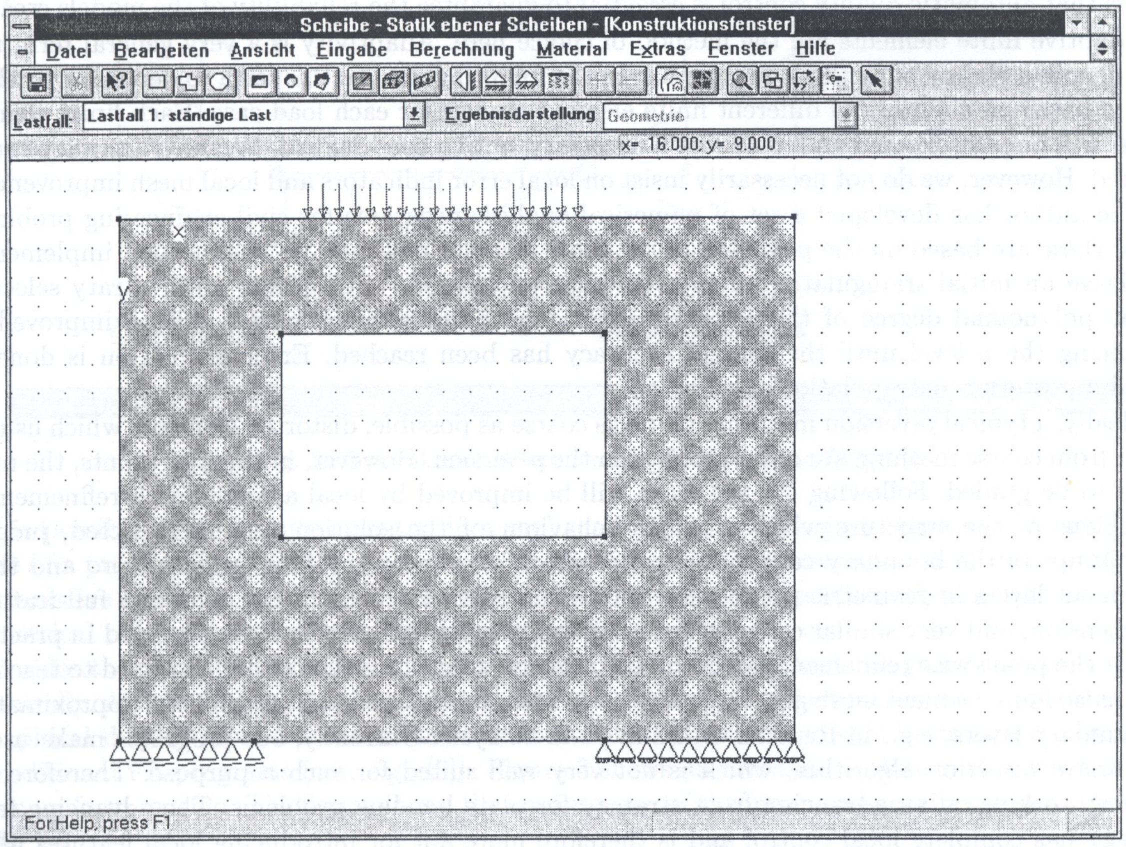
However, this kind of data import does not help much: The geometry needs to be simplified by omission of insignificant details like small holes, mid-surfaces and axes need to be constructed (not necessarily in the middle of the geometrical elements), and so on. These processes need to be guided by engineering expertise and experience, and therefore they are not very much amenable to software automation.

Another important problem when creating the structural model is dimensional reduction from the 3D reality to a 2D or 1D simplification. In this field, recent research indicates future chances for automatic procedures (see, e.g., [1] and [2]).

The amount of data required for a full description of the structural model is fairly small, compared to the data necessary to give a full geometrical description of the structure, including all details. On the other hand, the density of technical knowledge is high in the data pertaining to the structural model. Therefore, handling these data manually is easy and will not lead to confusion. Therefore, the user interface ought to give the user direct manipulation access to these data. The structural model does not contain any finite element discretization data yet. A modern software tool ought to separate these two kinds of models strictly in order to avoid confusion by a mess of finite element data. The traditional engineering education is focused on training engineers how to create the structural model from the real-world information, so that the average engineer is usually well educated for this job, as opposed to constructing a numerical discretization.

Data transmission from other programs plays only a marginal role here; at best, the engineer may at this time be aided by a "background image" of some CAD model of the structure. The information required for such a background image can easily be transferred by today's standard exchange formats, e.g., by the DXF format.

Figure 1 shows a typical glimpse of one of the author's structural analysis tools, here the plane elasticity program. The user works exclusively on the structural model, with no finite element information at all. The finite element discretization will be introduced only in the next step.



Lastfall: Lastfall 1: ständige Last

Ergebnisdarstellung: Geometrie

Fig. 1. Numerical analysis of plane elastic structures: Interface for interactive construction of the structural model

2.2. Automatical generation of the numerical analysis model

The structural model determines the mathematical solution of the structural analysis problem uniquely. Usually, an approximation to the exact solution will be obtained by a discretizing method like finite elements.

Very much like the structural model, a finite element model is also a geometrical and topological model of the structure. However, as opposed to the structural model, the numerical model is suited for automatical "mesh" generation, because it is not an independent model of ist own right, but only a formal discretization. Not only the topology and geometry can be mapped automatically, but also the boundary conditions and loads.

This way, a strategy evolves where the structural model is essential, and the user is never really confronted with finite element data and data manipulation belonging to finite element discretizations. This means also that the right way to input loads and support conditions is certainly not by "nodal forces" and "nodal constraints"; today's analyses are more often than not hampered by wrong finite element modelling of distributed loads and support conditions by inexperienced users fuzzing around with "nodal conditions".

Civil engineering structures are usually subject to a variety of different load cases. Generally speaking, each of these load cases defined for the same structure needs to be analyzed by an individual, separate numerical discretization. This is another strong hint that a practical tool needs to keep another, more comprehensive model of the structure than just an individual finite element model. The link between all these different finite element models is the unique structural model.

Creating the finite element model in an entirely automatical fashion out of the structural model means that automatic quality control is essential to guarantee the reliablility of the models created.

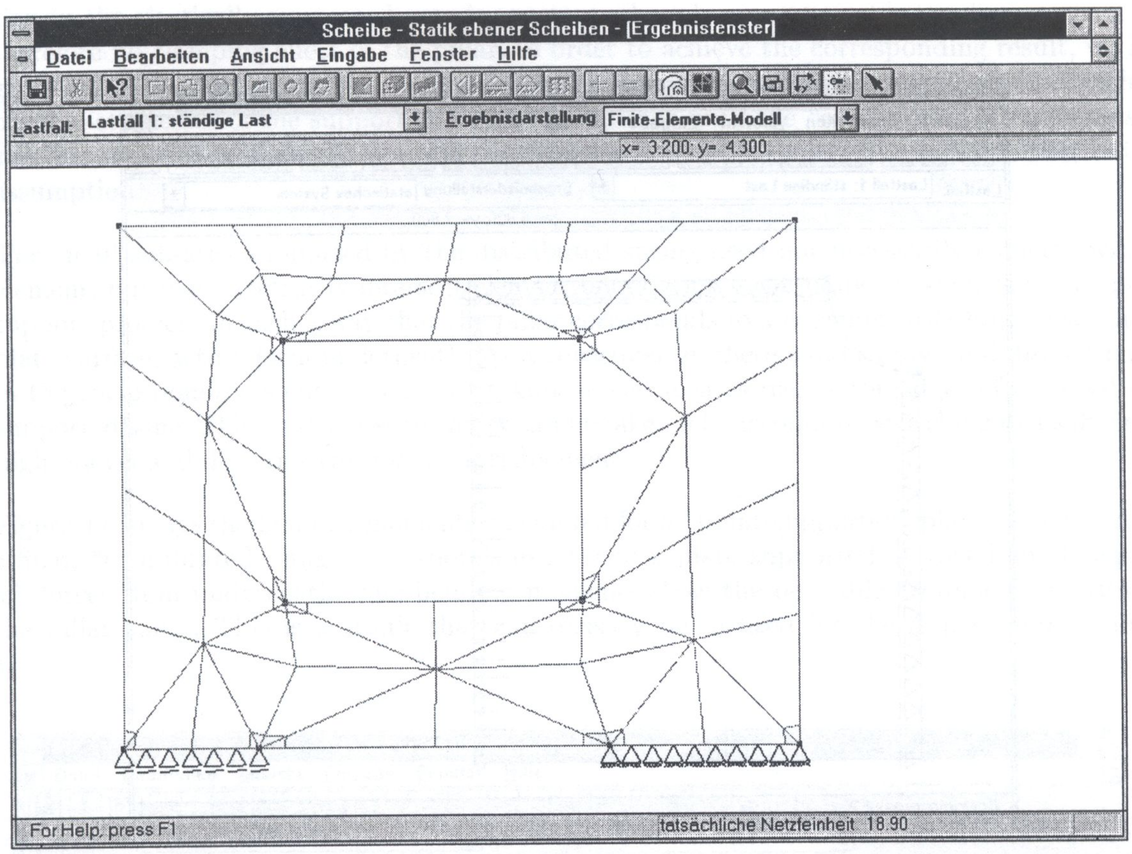
Adaptive finite elements are the method of choice here. Adaptivity is a very general term that may denote a variety of methods in practice. In the following, we use this term to denote an analysis that is based on a variety of different finite element models for each load case. Each finite element model will be post-checked and improved if necessary, until a user-defined threshold accuracy can be assured. However, we do not necessarily insist on local error indicators and local mesh improvement.

The author has developed a set of numerical analysis programs for civil engineering problems. All of these are based on the p-version of the finite element method [3]. The method implemented will leave an initial triangulation of the structure unchanged, but permits an arbitrary selection of the polynomial degree of the approximation. The finite element model will be improved by enhancing the p level until the desired accuracy has been reached. Error estimation is done by global a posteriori extrapolation.

Ideally, a typical p-version mesh should be as coarse as possible; distorted elements which usually result from coarse meshing are not a problem in the p-version. However, in singular points, the mesh needs to be graded. Following [3], the mesh will be improved by local a priori mesh refinement in subregions of the structure where singular behaviour of the solution can be expected, judging from jumps in the boundary conditions and re-entrant corners. Anything between zero and three refinement layers in geometric progression will be introduced. This strategy mimics a full-featured hp extension, and very similar convergence rates to a real hp extension can be observed in practice. Whilst the point-wise refinement can easily be implemented, it is not so straightforward to teach an automatic finite element mesh generator how to create a mesh that is suitable for the approximation of boundary layers, e.g., in Reissner–Mindlin plate analysis. Currently, our programs make use of a recursive bisection algorithm, which is not very well suited for such a purpose. Therefore, we are now working on an advancing front strategy for plate bending problems. The advancing front strategy has complete local control and is therefore more apt for introducing local features like a boundary refinement layer.

The plate bending and plane elasticity programs written by the author of the present paper start by creating a very coarse, uniform triangular mesh. Following an idea presented in [4], the triangles are then transformed into quadrilaterals. Quadrilateral elements are much better suited for plate analysis and are therefore preferred. After the triangle to quadrilateral transformation, the corner mesh refinements are introduced.

After the mesh construction, the user has the choice between various approaches. Either, he selects an "adaptive" approach in the sense of a global, uniform p-enrichment, or he decides to compute the structure only once, with a fixed polynomial degree, and to do another computation with a higher degree afterwards only if he finds that simple manual result checks such as equilibrium tests in a patch indicate insufficient accuracy. Currently, the program does not provide a residual- and jump-based local error indicator. Local effects like local distributed surface loads are taken into account heuristically by locally increasing the p-level. Typically, p varies between 5 and 8 for a plate computation. By selecting a more or less uniform p-distribution, the program can be implemented in a very efficient way, so that an immediate feedback to graphical interaction by the



Scheibe - Statik ebener Scheiben - [Ergebnisfenster]

Lastfall: Lastfall 1: ständige Last

Ergebnisdarstellung: Finite-Elemente-Modell

x= 3.200; y= 4.300

For Help, press F1

tatsächliche Netzfeinheit 18.90

Fig. 2. Typical mesh design for the *p*-version of the FEM

user is possible. This is quite essential in a tool that will be used, e.g., in the preliminary design phase of a project.Figure 2 displays a typical mesh design for the *p*-version of the FEM, created by the recursive bisection procedure, followed by the triangular to quadrilateral transformation and geometrical refinement at all possibly singular points.

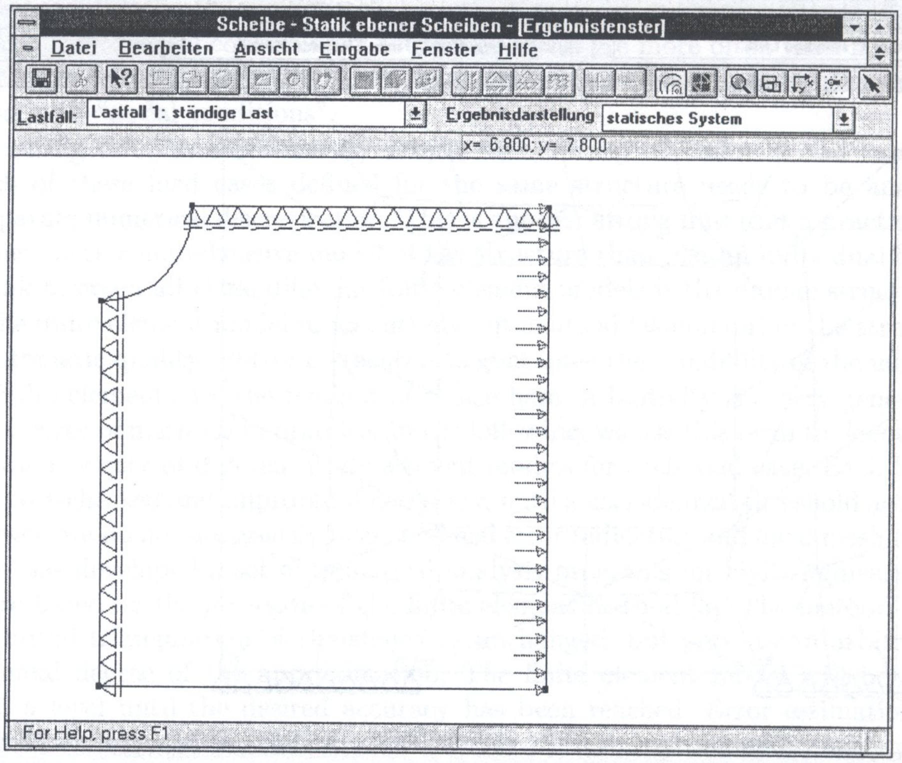
Besides the global error control by extrapolation, the user may check the pointwise convergence graphically. Although there is no mathematical proof available for pointwise stress convergence, such a graphical display gives a lot more information than the usual "one shot" evaluation of stresses in traditional FE programs (see, e.g., [5]). Figure 3 shows an example, a plate with a circular hole under uniform tension. The figure shows the convergence of the maximum tensile stress at the edge of the hole.

The plate bending program described here offers a unique model for point-supported plates: Point support is illegal in Reissner–Mindlin (shear deformable) plate theory; however, many practical problems such as pile-supported foundations or plates supported by pillars call for some model for "point-support". Traditional FE codes offer no assistance in this field. We have implemented the following strategy:

- The pillar will be replaced by a distributed spring in a small area. The spring stiffness can be obtained from the actual stiffness of the pillar or pile.

- In case of a moment-free support, this model will not render the correct answer because, in the general case, the plate will rotate in the spring-supported area; this rotation will create an uneven distribution of the stresses in the springs, and the spring support will introduce a resulting bending moment in addition to the supporting normal force. Therefore, the influence of the pillar will be over-estimated.

- To avoid this problem, we introduce an additional assumption: Provided the stress distribu-


Scheibe - Statik ebener Scheiben - [Ergebnisfenster]
Datei Bearbeiten Ansicht Eingabe Fenster Hilfe
Lastfall: Lastfall 1: ständige Last
Ergebnisdarstellung statisches System
x= 6.800; y= 7.800
For Help, press F1

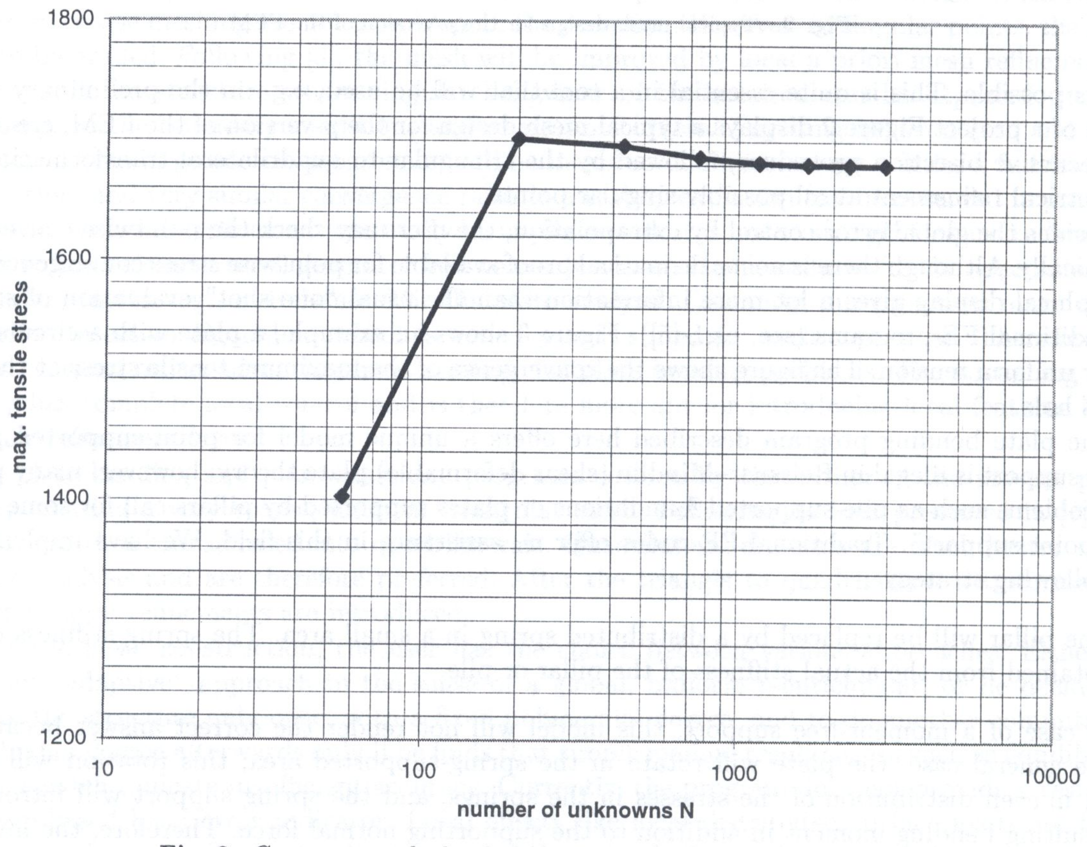
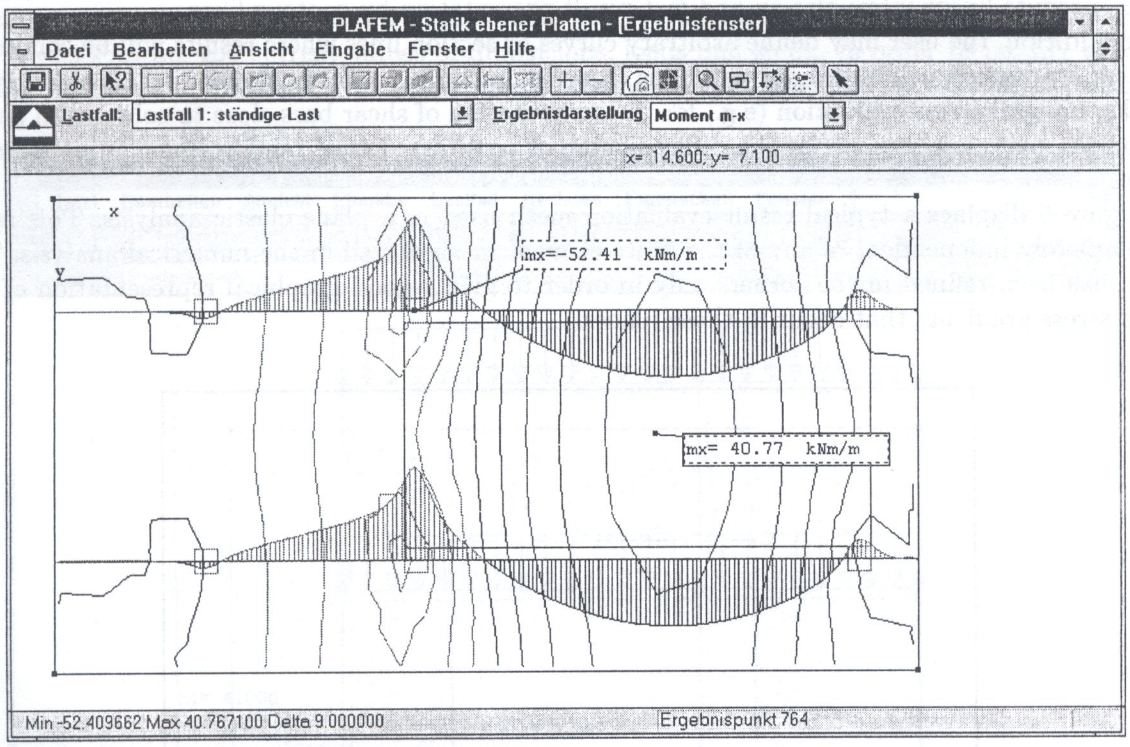


Fig. 3. Convergence of a local, design-critical result in the *p*-version

tion in the elastically supported area is constant, there is no undesirable bending moment and therefore no clamping effect of the pillar. In order to achieve the corresponding result, we integrate the "out-of-balance" bending moments and perform an "equilibrium" iteration with the original stiffnesses of the supported plate. We continue until the "out-of-balance" moments are small enough. At this stage, the results correspond exactly to the "constant stress distribution" assumption.

- The small sub-area supported by the distributed spring need not necessarily coincide with an element, but may arbitrarily intersect element boundaries. Eventually, we shall solve the point-support problem in such a way that the pillar corresponds to a constant distributed load on the plate surface, acting form underneath. As a consequence, there will be only mild discontinuities in the corresponding solution, namely, a kink in the shear force, at the edges of the elastically supported zone. This kind of discontinuity can be taken into account by switching to a sufficiently high p-level and does not call for mesh refinement.

Figure 4 displays the bending moments calculated for a "point-supported" plate by this method. The final, "equilibrated" stage corresponds exactly to a plate supported by distributed "equilibrium" forces from underneath. The bending moments show the desirable "rounded" distribution in the pillar zones. This is a result that can directly be utilized for the dimensioning and design.


PLAFEM - Statik ebener Platten - [Ergebnisfenster]

Lastfall: Lastfall 1: ständige Last

Ergebnisdarstellung: Moment m-x

x= 14.600; y= 7.100

Min -52.409662 Max 40.767100 Delta 9.000000

Ergebnispunkt 764

Fig. 4. Analysis of a point-supported plate. The plate rests on six quadratic pillars. Bending moment m_x

The p-version of the FEM turns out to be very effective for engineering analyses of plate structures. E.g., stiffened plates can be modeled by combining Timoshenko beams and Reissner–Mindlin plate elements. This is definitely a better model than the usual nodal coupling between the plate model and Bernoulli beam elements in traditional, h-version FEM with low polynomial degrees.

As opposed to the structural model, the finite element discretization contains a lot of data; however, its content of original, new technical information that needs to be handed over to other processes in the building design is comparatively low; therefore, it is best to view the finite element model only as a transient one that may be discarded immediately after usage. As it can be constructed automatically by a set of rules, it is not very effective to leave its design to the user, who may be quite inexperienced in numerical methods. Also, it is not very reasonable to view the final design-critical results as attributes to a specific finite element mesh since each load case may require an individual finite element discretization in order to reach a prescribed accuracy level, and each of these finite element models may be discarded after the numbers have been gained. Rather, a unique result mesh is required.

2.3. Result evaluation and superposition

In our program, result evaluation will be primarily based on an arbitrarily designed and arbitrarily fine result evaluation and data storage mesh that is independent of any special finite element model constructed for the sake of numerical analysis. Whereas the finite element models for all the individual load cases are viewed as transient data, the data mesh is kept for further processing. Apart from being used as a basis for graphical result display and quick evaluation, it also serves for the superposition of load cases and worst-case analysis, mapping for remeshing in the analysis of construction processes where the structure changes in time, and so on.

In order to create this result and data mesh, any mesh generator may be employed. The resulting "elements" may be rather arbitrarily shaped, and it is most convenient to use a triangular meshing which permits linear interpolation and fast result presentation by contour lines.

In addition, the user may define arbitrary curves or section lines where results will be computed and superpositions will be carried out. This aids the user in plausibility checks such as equilibrium checks, integral stress evaluation (e.g., for the computation of shear bars close to columns in point-supported plate analysis, or for the dimensioning of prestressed steel reinforcement in prestressed plates).

Figure 5 displays a typical result evaluation mesh used in a plane elastic analysis. This mesh is completely independent of any of the finite element meshes used in the numerical analysis. The mesh has been refined in the corners only in order to give a good graphical representation of the steep stress gradients that occur in those places.

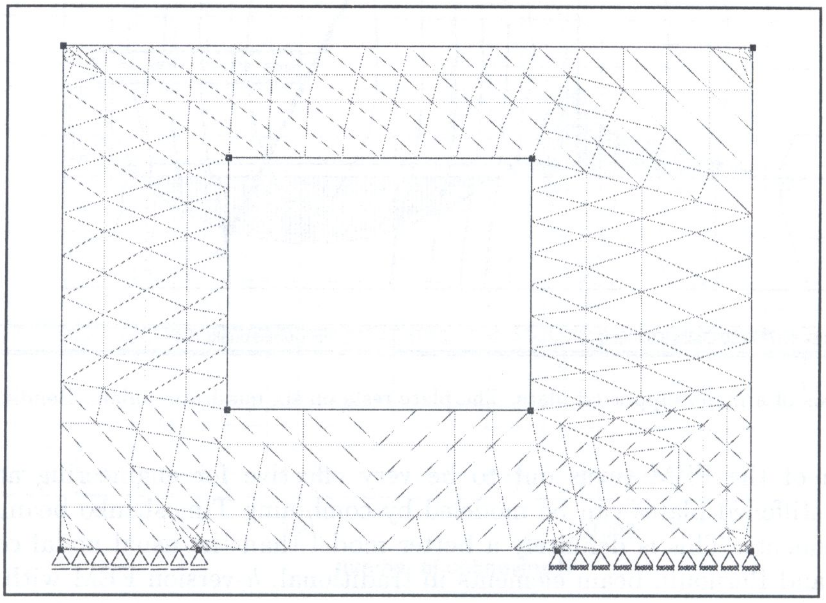


Fig. 5. Typical result evaluation mesh in the analysis of a slab

2.4. Graphical interactive user environment

An essential part of any program to be used in industrial practice is the user interface. There is no way without a graphical interactive user environment.

The author of the present paper has experimented with various approaches for graphical interactive user environments.The most promising way to design user–program interaction has turned out to be the "object-oriented" technique: In this approach, the user selects the object he wants to operate on first and specifies the action to be performed next. The action may alternatively be specified implicitly by mouse movement or key pressing.

Such an approach is more difficult to implement than the traditional "predicate-oriented" approach where the action to be carried out has to be selected first, and where the user has to specify the type of object which this action should be performed on, prior to selecting the actual object. This "predicate-oriented" approach leads quickly to huge menu trees, and users view this approach as roundabout and little "natural". Therefore, the author has switched from this way of user-interaction to the "object-oriented" type.

It is well worth looking around in the world of CAD to find efficient models for user–program interaction even in numerical analysis. A very effective idea is parametric design. Figure 6 shows a typical situation in the author's p-version based beam finite element analysis program. Frequently, similar structures like the multi-storeyed frame displayed in the figure occur frequently. In our approach, every dimension and every load and material property of the structure shown can be assigned a name or a formula. That way, the user can prepare a "macro" or template of his own for frequently occuring types of structures, and in any special analysis, he simply clicks on the text labels showing the current values of dimensions, loads and so on, in order to modify the predefined structure according to his actual needs. Instead of purchasing a set of special programs

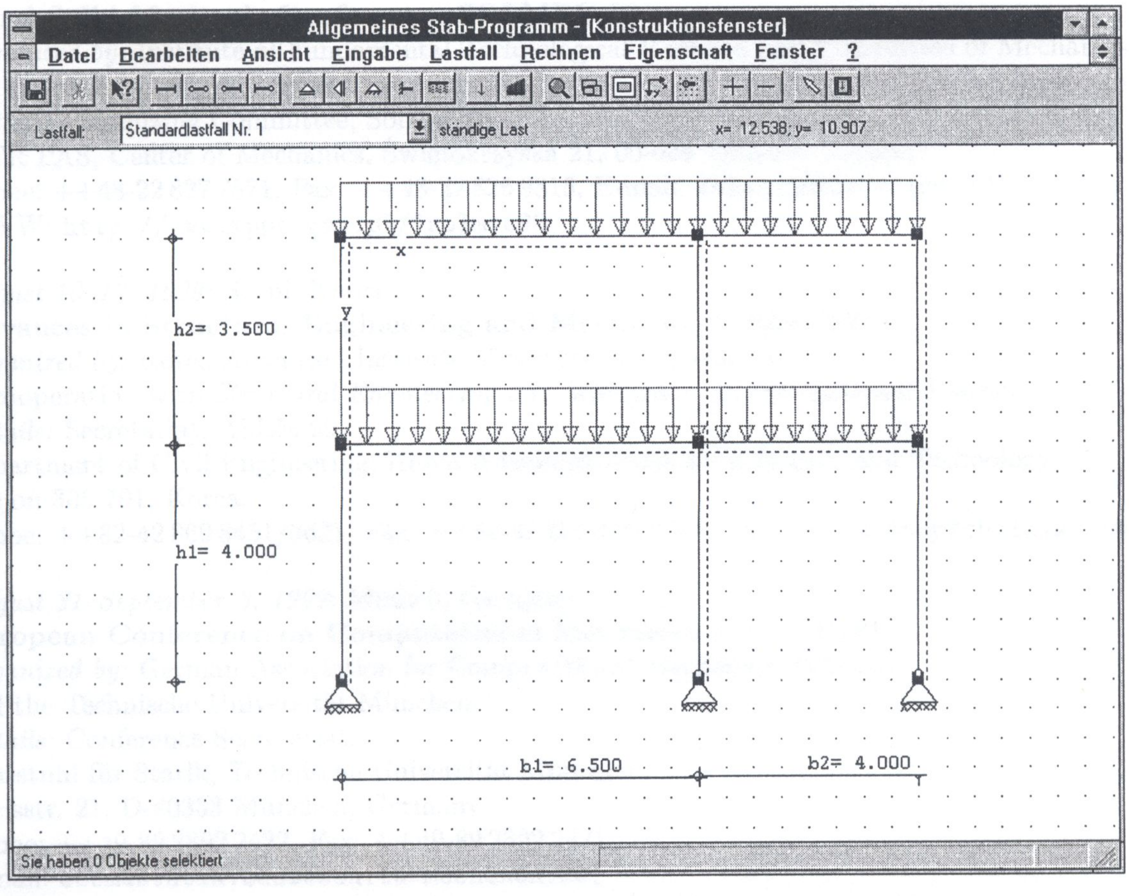

Allgemeines Stab-Programm - [Konstruktionsfenster]
Datei Bearbeiten Ansicht Eingabe Lastfall Rechnen Eigenschaft Fenster ?
Lastfall: Standardlastfall Nr. 1
ständige Last
x= 12.538; y= 10.907
x
y
h2= 3.500
h1= 4.000
b1= 6.500
b2= 4.000
Sie haben 0 Objekte selektiert

Fig. 6. Working with parametrized templates: multi-storeyed frame defined by four parameters

for various simple problems, the users themselves may this way generate a family of tailor-made software products that match their individual needs perfectly.

3. Conclusion

Advanced numerical methods offer a real chance for a revival of traditional engineering methods. The engineer can interact with software based on automatical mesh generation, error control and flexible, individual result evaluation, the same way he uses the traditional handbook. Designing the structural model rather than dealing with details of the numerical discretization, the user gains a lot of overview and insight and is enabled to control the numerical analysis process by the engineering criteria he has been taught in his education, rather than feeling subject to a black-box uncontrollable analysis system.

References

[1] E. Stein, S. Ohnimus. Dimensional adaptivity in linear elasticity with hierarchical test spaces for h- and p-refinement processes. *Engrg. Comp.*, **12**:107–119, (1996).

[2] B.A. Szabó, G. Sahrman. Hierarchic plate and shell models based on p-extension. *Int. J. Num. Meth. Engrg.*, **26**:1855–1881, (1988).

[3] B.A. Szabó, I. Babuska. *Finite Element Analysis.* John Wiley & Sons, New York (1991).

[4] E. Rank, M. Schweingruber, M. Sommer. Adaptive mesh generation and transformation of triangular to quadrilateral meshes. *Comm. Appl. Num. Meth. Engrg.*, **9**:121–129, (1993).

[5] Z. Yosibash, B.A. Szabó. Convergence of stress maxima in fifnite element computations. *Comm. Num. Meth. Engrg.*, **10**:683–697, (1994).