

Der folgende Beitrag erschien 1997 in "Bauingenieur-Bauinformatik", Heft 3, März 1997.

Gestaltung ingenieurgemäßer Statiksoftware

S. M. Holzer¹

Zusammenfassung Im Zusammenwirken von komfortabler graphisch-interaktiver Handhabung, standardisierter Gestaltung der Oberfläche und fortschrittlicher Finite-Elemente-Technologie können praxisorientierte Statikprogramme entwickelt werden, die es dem Benutzer gestatten, mit traditionellen Denkmodellen und Arbeitstechniken zu arbeiten, zu jeder Zeit den vollen Überblick zu behalten, und dennoch die Flexibilität und Leistungsfähigkeit moderner Berechnungsmethoden voll auszuschöpfen. Strategien und Techniken werden anhand konkreter Beispiele erläutert.

Designing structural analysis software to match the needs of engineers

***Abstract** New developments in structural analysis methods such as automatic error estimation, adaptive discretization control, and mesh generation penetrate the engineering practice slowly. 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 Integration des Rechners in die Arbeit des Statikers

Rechner dienen in der täglichen Arbeit des Statikers traditionell als reines Rechenwerkzeug. Mit der allgemeinen Durchdringung der Arbeitswelt durch immer preiswertere und leistungsfähigere Rechner - direkt auf dem Schreibtisch jedes einzelnen - verlagern sich jedoch immer mehr Tätigkeiten, die bisher mit Papier und Bleistift ausgeführt worden sind, ebenfalls auf den Rechner - weit über reine Berechnungstätigkeiten hinaus. Der Computer - in der Praxis der Ingenieurbüros und Baufirmen überwiegend der PC - hat sich als leistungsfähiges Allzweckwerkzeug, gerade auch bei nicht numerischen Aufgabenstellungen, durchgesetzt (Textverarbeitung, Datenbanken). Die rasante Entwicklung nicht nur der Rechnerleistung, sondern auch der Technologie der Peripheriegeräte (hochauflösende Bildschirme, Laserdrucker, Kommunikationstechnologie) hat überdies die Ansprüche der Kunden und Geschäftspartner ansteigen lassen, so daß professionelle Graphiken, aber auch jegliche Texte, z.B. Statische Berichte, heute vielfach auf dem PC erstellt werden müssen, um die Erwartungen aller Beteiligten zu genügen.

Mit der Durchdringung der Arbeitswelt durch Rechner auf breiter Front tritt nunmehr das Problem in den Vordergrund, die vielen unabhängig voneinander entwickelten und durchaus nicht ohne weiteres zusammenpassenden computerbasierten Werkzeuge in ein einheitliches, übergreifendes Gesamtkonzept einzubinden ("Integration"). Bei bisherigen Analysen und Entwicklungsanstrengungen stand meist die Idee des Datenaustausches zwischen unterschiedlichen Programmen verschiedener Hersteller im Vordergrund [1]. Leitgedanke war dabei, Kosten und Fehler zu vermeiden, die durch mehrfaches Eingeben derselben Information entstehen können.

Ein verbesserter Datenfluß zwischen verschiedenen Softwareprodukten für Teilaufgaben ist aber nur einer von vielen Teilaspekten eines umfassenden Konzepts zur Effizienzsteigerung und Qualitätssicherung.

Ein anderer, ebenso bedeutsamer Aspekt ist die Vereinheitlichung der Handhabung unterschiedlichster Programme. Der Durchschnittsbenutzer arbeitet nicht andauernd und tagtäglich mit demselben Programm, sondern wechselt zwischen unterschiedlichsten Applikationen. Wenn der Benutzer sich dabei jeweils in einer anders gestalteten Umgebung zurechtfinden und mit einer anderen Eingabephilosophie auseinandersetzen muß, sind Fehler und langsames, ineffizientes Arbeiten vorprogrammiert.

Neben der Einheitlichkeit der Bedienung ist eine Benutzer-Programm-Interaktion dringend erwünscht die dem Benutzer die Gelegenheit gibt, seine Vorstellungen in ingenieurgemäßen Begriffen und Darstellungen an den Rechner weiterzugeben und umgekehrt Ergebnisse in ingenieurgemäßer Aufbereitung auszuwerten, so daß z.B. manuelle Kontrollrechnungen oder Plausibilitätsprüfungen erleichtert werden [2].

Der folgende Aufsatz umreißt Chancen, wie neue Statikprogramme durch eine Verbindung moderner Entwicklungen der Berechnungsmethoden und der Techniken der graphischen Interaktion den geschilderten Zielen wesentlich näher gebracht werden können.

2 Design neuer Statik-Software

Die Tätigkeit des Statikers besteht darin, die Tragwirkung des Bauwerkes durch ein statisches System zu beschreiben, anhand dieses statischen Systems eine numerische Berechnung durchzuführen und zum Schluß die Berechnungsergebnisse systematisch auszuwerten, um diejenigen Daten zu identifizieren, die für die Konstruktion des Bauwerkes ausschlaggebend sind. Aus diesen Daten entwickelt der Statiker einen Konstruktionsvorschlag.

Zu jedem dieser drei Schritte gehören spezifische Modelle der Baustruktur:

1. statische Modelle
2. numerische Modelle
3. Ergebnismodelle

Diese drei Modelle ähneln sich, oberflächlich betrachtet, darin, daß sie alle auf einer geometrischen Beschreibung der Baustruktur beruhen. Dies darf jedoch über die grundlegenden Unterschiede zwischen diesen Modellen nicht hinwegtäuschen. Im statischen Modell (1) sind wesentliche Idealisierungsannahmen über das Tragverhalten der Baustruktur verankert, während die numerischen Modelle (2) lediglich geometrische Hilfskonstruktionen zur Durchführung einer numerischen Berechnung darstellen und keineswegs unabhängig vom statischen Modell gewählt werden können. Auch die Ergebnismodelle (3) sind lediglich Hilfskonstruktionen zur Auswertung der Berechnungsergebnisse. In der Vergangenheit sind die Unterschiede zwischen diesen drei Modelltypen nicht klar genug herausgestellt worden. Daher sind konventionelle Finite-Elemente-Programme so strukturiert, daß alle Daten als Attribute zum numerischen Modell (d.h. an der Finite-Elemente-Topologie) gespeichert werden, sowohl die Informationen des statischen Modells als auch die Ergebnisdaten.

Bei jedem Modellierungsschritt werden neue Vereinfachungen eingeführt. Diese Modellierungsfehler können und müssen kontrolliert werden, um zuverlässige Ergebnisse zu erhalten. Keines der genannten drei Modelle ist eindeutig festgelegt, sondern seine Wahl muß sorgfältig bedacht werden, und die Qualität und Verlässlichkeit der statischen Analyse hängt entscheidend von der adäquaten Wahl der jeweiligen Modelle ab.

Ziel moderner Berechnungsprogramme muß es daher sein, dem Statiker größtmögliche Flexibilität bei der Wahl der jeweiligen Modelle zu geben und ihm eine möglichst einfache Handhabung dieser

Modelle zu gestatten. Im folgenden werden die Zielvorstellungen umrissen, denen ein modernes Statikprogramm folgen sollte. Beispiele aus Programmen des Verfassers, die dieser bei HOCHTIEF in Frankfurt entwickelt hat, demonstrieren die Machbarkeit und Leistungsfähigkeit der vorgeschlagenen Ansätze.

2.1 Automatisierte Bildung des statischen Systems

Im ersten Modellierungsschritt abstrahiert der Statiker die physikalische Realität (Geometrie, Materialverhalten, Lasteinwirkungen) und entwickelt ein *statisches Modell* (statisches System mit Geometrie-, Material-, Lastfall- und Lastinformationen, Randbedingungen). Das statische Modell ist eine rein mathematische Beschreibung der Realität.

Grundlage des statischen Modells ist eine vereinfachte Geometrie und Topologie des Bauwerkes. Die Frage ist nun, inwieweit der Ingenieur bei der Bildung des statischen Systems durch Übernahme von Daten aus anderen Bauwerksmodellen - z.B. aus CAD-Programmen - und durch Automatismen unterstützt werden kann:

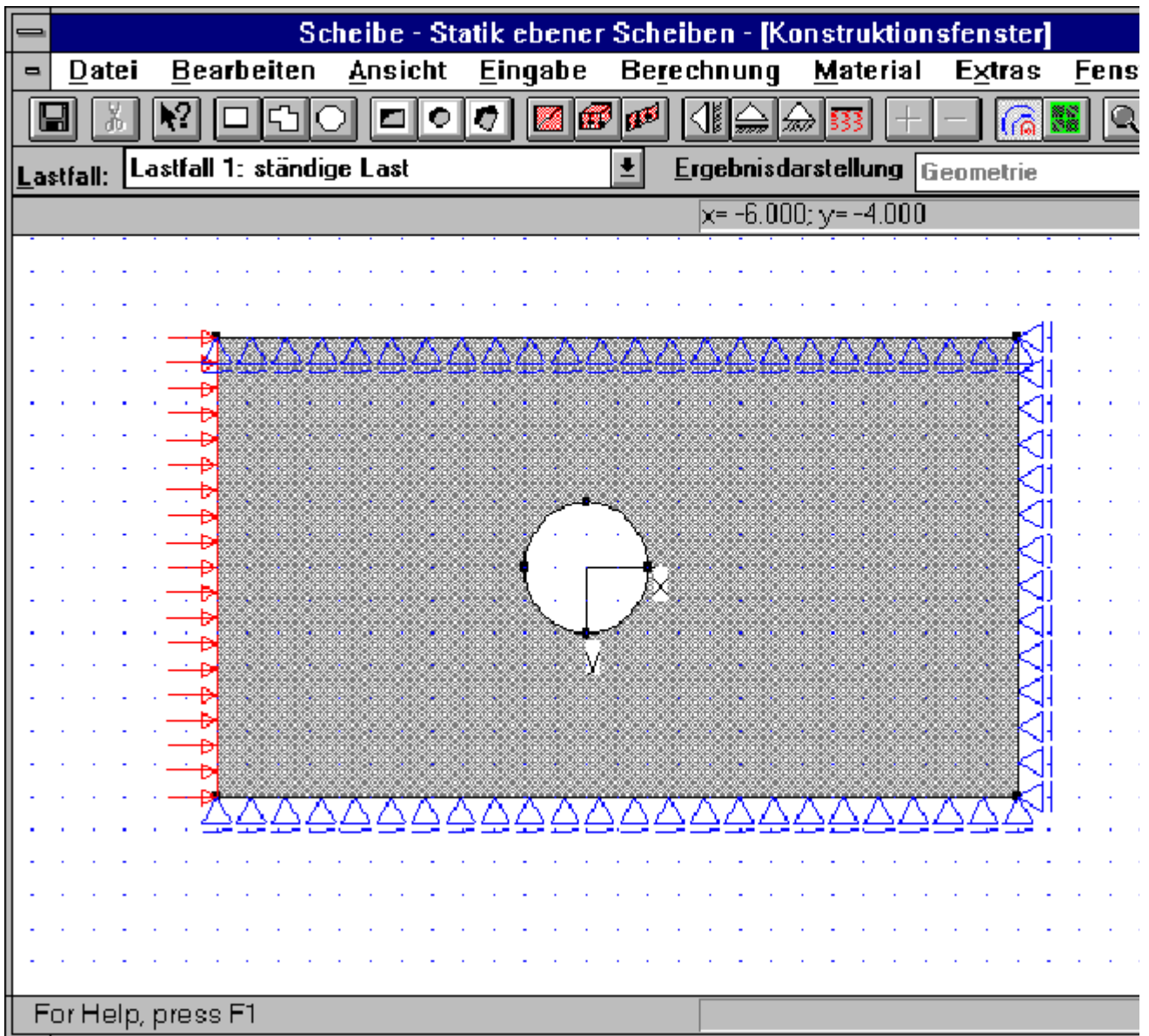
Schon die Ableitung der Geometrie des statischen Systems (Weglassen unwichtiger Details, z.B. kleiner Aussparungen, Lage der Systemlinien und Auflagerachsen usw.) aus der realen Bauwerksgeometrie ist alles andere als trivial; Ingenieurverstand und Erfahrung sind unerlässlich, um auf diesem Sektor verlässliche Entscheidungen zu fällen. Daher kann schon dieser Schritt kaum je vollständig automatisiert werden.

Ein weiteres Problem bei der Erzeugung des statischen Systems ist die Dimensionsreduktion vom vollen 3D-Körper, d.h. die Wahl zwischen verschiedenen Idealisierungen, ausgehend vom 3D-Kontinuum. Die gleiche Struktur könnte beispielsweise als Platte nach einer höheren oder einfachen Plattentheorie oder als Balken idealisiert werden. In diesem Bereich sind im Moment Forschungsarbeiten im Gange. In Zukunft kann sich vielleicht ein Berechnungsprogramm wenigstens halbautomatisch zum Übergang auf einem stärker an der 3D-Wirklichkeit Modell entscheiden, falls dies in lokalen Bereichen, z.B. Stützenbereichen, sinnvoll ist ([3], [4]). Der Verfasser selbst arbeitet zur Zeit an Finite-Elemente-Berechnungsmethoden für Platten, die auf einem vollen 3D-Modell beruhen, aber in Dickenrichtung der Platte mit einfachen Ansätzen arbeiten (vgl. auch [4]), um die Datenmenge des Finite-Elemente-Modells zu beschränken.

Die Datenmenge, die das statische System beschreibt, ist vergleichsweise gering, der Zugewinn an technischer Information, der in dieser kleinen Datenmenge steckt, ist jedoch bedeutend, die Informationsdichte der Daten also hoch. Solche Daten sind prädestiniert zur direkten Manipulation durch den Statiker in einem modernen Statik-Programm. Das statische System enthält noch keinerlei Finite-Elemente-Informationen. Ein modernes Baustatikprogramm sollte sich auf den Dialog des Benutzers mit dem Programm auf Basis des statischen Systems konzentrieren, da in dieser Stufe der Modellierung die ingenieurmäßigen Begriffe und Vorstellungen im Vordergrund stehen. Der Statiker ist speziell für diese Aufgabe ausgebildet.

Die Bedeutung der Übernahme von Daten aus anderen Programmen, z.B. einem CAD-Paket, spielt hier eine untergeordnete Rolle, so daß das Problem der Datenschnittstelle in den Hintergrund tritt. Viel mehr Unterstützung als durch die Übernahme von geometrischer Information als "Hintergrundbild", das den Statiker bei der Konstruktion des statischen Systems visuell unterstützt, ist nicht zu erwarten. Um solche Informationen in das Statik-Programm zu laden, sind einfache, bereits heute übliche Austauschformate wie das DXF-Format ausreichend.

Fig. 1 zeigt einen Ausschnitt aus der Benutzeroberfläche des Scheibenprogramms des Verfassers. Der Benutzer bearbeitet ausschließlich das statische System, unbelastet durch irgendwelche Finite-Elemente-Informationen, die erst im nächsten Modellierungsschritt ins Spiel kommen.



(Fig. 1) Scheibenprogramm: Graphisch-interaktive Erstellung des statischen Systems. - Numerical analysis program: Interface for interactive construction of the structural model.

2.2 Automatisierte Bildung des numerischen Modells aus dem statischen System

Soweit das durch das statische Modell gegebene mathematische Problem eindeutig lösbar ist, ist die mathematische Lösung des Problems mit der Aufstellung des Strukturmodells eindeutig festgelegt. Die Gewißheit der Existenz und Eindeutigkeit einer exakten Lösung des mathematischen Problems garantiert natürlich noch lange nicht, daß diese Lösung auch mit endlichem Aufwand gefunden werden kann oder gar analytisch ist. In der Regel wird versucht, eine möglichst gute Näherungslösung mit Hilfe der Methode der finiten Elemente zu finden, also mit Hilfe eines numerischen Modells.

Solche numerischen Modelle sind ebenfalls topologisch-geometrische Modelle des Bauwerkes. Im Gegensatz zu der Geometrie des statischen Systems kann die Geometrie eines Finite-Elemente-Modells jedoch systematisch und automatisch aus der des statischen Modells entwickelt und erstellt werden. Dies leisten automatische "Freivernetzer".

Alle anderen Informationen wie Randbedingungen und Lasten können danach ebenfalls streng nach Schema und somit automatisch von einem statischen Modell auf ein Finite-Elemente-Modell abgebildet werden.

Das Statikprogramm der Zukunft muß neben der bloßen Generierung der Netztopologie auch diese Abbildung der vom Benutzer anhand des statischen Systems spezifizierten Randbedingungen und Lasten leisten. Bei der Berechnung elastischer Kontinua bedeutet dies unter anderem:

- Der Benutzer gibt Randbedingungen auf keinen Fall als "Knotenfesthaltungen" oder "Knotenlasten" ein; vielmehr werden Festhaltungen kantenweise und Lasten als Rand-, Linien- oder Flächenlasten, unabhängig vom Finite-Elemente-Netz und ohne Kenntnis desselben eingegeben. Ein automatischer Prozeß hat dafür zu sorgen, daß diese Informationen adäquat auf das Finite-Elemente-Netz übertragen ("diskretisiert") werden. Viele Fehler in Finite-Elemente-Berechnungen entstehen durch falsche Diskretisierung von Randbedingungen und Lasten durch unzureichend mit der Finite-Elemente-Theorie vertraute Benutzer. Diese Fehler sind ausnahmslos vermeidbar.
- Das Programm verhindert automatisch *unzulässige Eingaben* wie punktförmige Festhaltungen oder Punktlasten. Es kann nicht oft genug betont werden, daß Knotenfesthaltungen bei Finite-Elemente-Modellen von Scheiben oder Platten nach der Mindlin-Theorie nur paarweise (als diskretisierte Kantenfesthaltung) auftreten dürfen und daß Knotenlasten *keine* Punktlasten sind, sondern einer gewissen verteilten Last entsprechen. Anstatt die unterstellte Lastverteilung, die einer Knotenlast entspricht, von dem "beliebig" wählbaren Finite-Elemente-Modell abhängig zu machen, sollte der Statiker genau angeben, wie er sich die Lastverteilung vorstellt. Beispielsweise können vermeintliche "Punktlasten" durch verteilte Lasten auf einer kleinen Lastfläche beschrieben werden. Das Finite-Elemente-Programm hat sich dann darum zu kümmern, daß in der Umgebung derartiger lokaler Effekte ein adäquates Finite-Elemente-Modell gewählt wird (nicht notwendigerweise eine Netzverfeinerung).

Für verschiedene Lastfälle sind in aller Regel verschiedene Finite-Elemente-Modelle erforderlich, um vergleichbare Ergebnisqualität zu erhalten. Ein entsprechendes Vorgehen kann man nur dann implementieren, wenn alle wesentlichen Systeminformationen in Form eines von Finite-Elemente-Daten unabhängigen statischen Modells beschrieben und gespeichert werden.

Ein Verfahren, das die Finite-Elemente-Diskretisierungen vollautomatisch aus dem statischen System erzeugt, setzt natürlich voraus, daß eine ausreichende Qualität der Finite-Elemente-Ergebnisse automatisch gewährleistet wird.

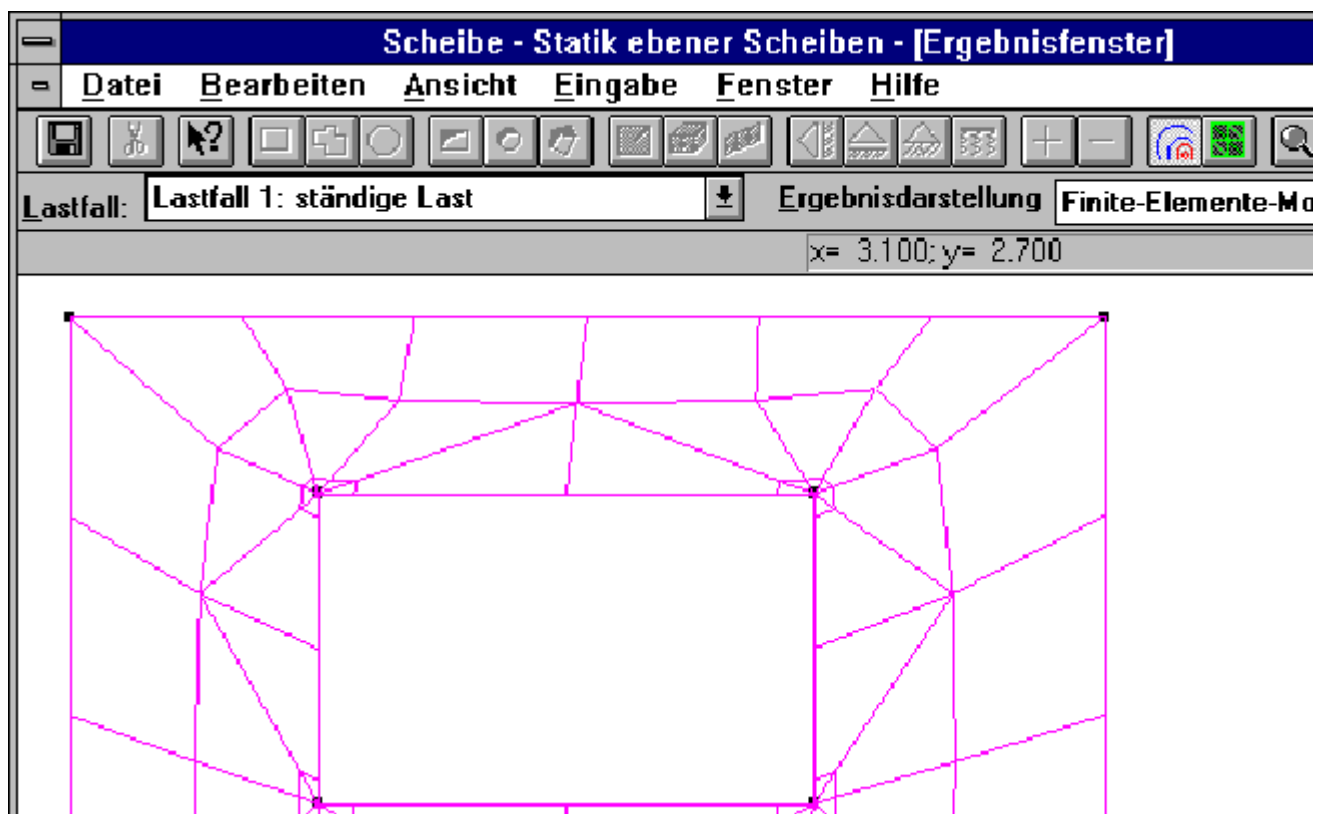
Dies leisten die adaptiven Berechnungsverfahren. Adaptivität im strengen Sinne bedeutet hierbei, daß die Berechnung in mehreren Durchläufen, mit unterschiedlichen Finite-Elemente-Diskretisierungen, durchgeführt wird und am Ende jedes Durchlaufes die Ergebnisgenauigkeit anhand lokaler Verfeinerungsindikatoren und globaler Genauigkeitsschätzungen überprüft wird. Entsprechend den lokalen Indikatoren kann das Finite-Elemente-Modell sodann lokal verbessert werden (z.B. durch Netzverfeinerung oder Ansatzgraderhöhung), um durch möglichst ökonomischen Einsatz der Ressourcen eine möglichst deutliche Verbesserung der globalen Genauigkeit zu erzielen. Von dieser "reinen Lehre" der Adaptivität - kann freilich in einem Programm für die Baupraxis durchaus in pragmatischer Weise abgewichen werden:

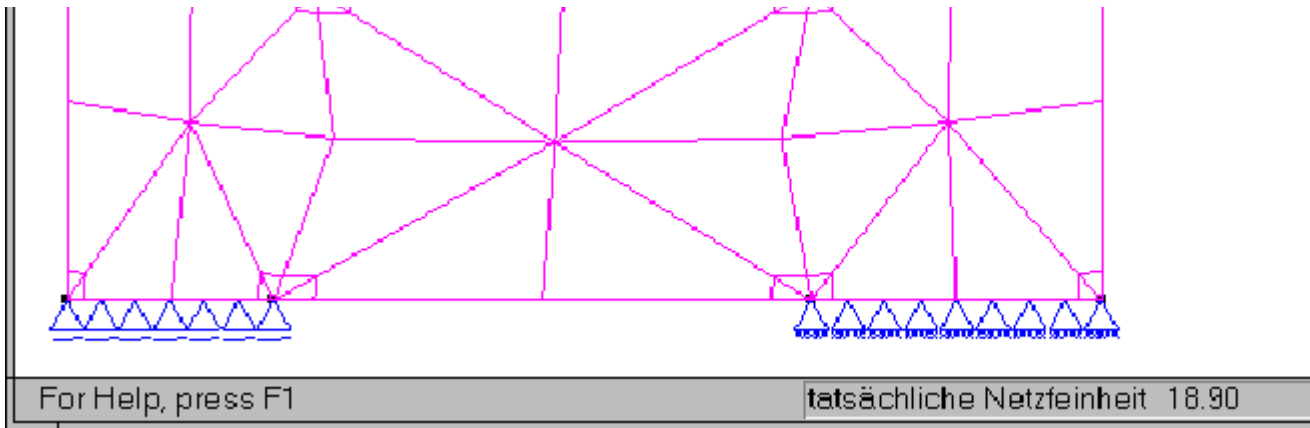
Die hier als Beispiel besprochenen Statikprogramme des Verfassers basieren auf der p -Version der Finite-Elemente-Methode [5]. Diese Methode geht von einer festen Netzeinteilung aus, jedoch ist der Polynomgrad des Ansatzes auf diesem festen Netz frei wählbar. Mißt man die Ergebnisqualität einer Finite-Elemente-Methode mit globalen, also über das gesamte Netz gemittelten Genauigkeitskriterien, und untersucht man den Zusammenhang zwischen der Gesamtanzahl der Koeffizienten einer Finite-Elemente-Lösung und der erzielten Genauigkeit, so kann man die verschiedenen Strategien zur Verbesserung von Finite-Elemente-Lösungen vergleichen. Dabei stellt sich heraus, daß in den Anwendungsfällen der Statik die Polynomgraderhöhung gegenüber der reinen Netzverfeinerung überlegen ist: Bei Aufgaben, deren exakte Lösung glatt ist, führt die Ansatzgraderhöhung der p -Methode zu einer exponentiellen Abnahme des globalen Fehlers, während die herkömmliche

gleichmäßige Netzverfeinerung nur zu algebraischer Konvergenz führt. Aber auch bei Problemen deren exakte Lösung singuläre Punkte (unendliche Spannungsspitzen) aufweist, ist die reine Polynomgraderhöhung der p -Version einer gleichmäßigen Netzverfeinerung noch überlegen [5]. Bei Problemen des letztgenannten Typs ist freilich eine gleichzeitige lokale Netzverfeinerung und Polynomgraderhöhung (hp -Version) erforderlich, um wiederum eine exponentielle Abnahme des Approximationsfehlers zu erzielen.

Ideale Netze für die p -Version sind möglichst grob, und verzerrte Elemente, wie sie beim Versuch der möglichst groben Vernetzung eines Gebietes oft entstehen, sind völlig unproblematisch. Will man auch bei Problemen mit singulären Punkten ein effizientes Rechenverfahren erzielen, so reicht es aus, an lokalen Spannungskonzentrationen zusätzlich zur Polynomgraderhöhung eine lokale Netzverfeinerung durchzuführen. Damit täuscht man eine echte hp -Methode vor und kann die Genauigkeit der Ergebnisse deutlich verbessern. Im Idealfall gelingt es mit einer solchen Methode, im praktisch relevanten Bereich der Konvergenzkurve nahezu der echten hp -Methode zu folgen; erst bei einer weiteren Erhöhung der Anzahl der Unbekannten über den praktisch sinnvollen Bereich hinaus stellt sich dann die in Wirklichkeit schlechtere Konvergenzcharakteristik einer reinen p -Methode ein.

Im Platten- und Scheibenprogramm des Verfassers wird zunächst ein möglichst grobes Dreiecksnetz für die Struktur erzeugt. Dieses Dreiecksnetz wird sodann in ein reines Vierecksnetz verwandelt [6], da Finite-Elemente-Netze aus Vierecken bessere Konvergenzeigenschaften besitzen. Anstatt die erforderlichen Netzverfeinerungen an Spannungskonzentrationen iterativ und adaptiv zu ermitteln, wird das Netz gleich zu Anfang an kritischen Punkten verfeinert. Solche Punkte können anhand einfacher geometrischer und mechanischer Kriterien auf Basis des statischen Systems ermittelt werden: Wechsel in der Auflagerbedingung, einspringende Ecken usw. Danach wird nur noch der Polynomgrad angepaßt. Der Benutzer kann entweder das Modell manuell mit verschiedenen Polynomgraden rechnen, den Polynomgrad automatisch schrittweise erhöhen lassen, bis die gewünschte Genauigkeit erreicht wird, oder einfach nur eine einzige Lösung mit einem sehr hohen Polynomgrad bestimmen. Auf die lokale Auswertung von Fehlerindikatoren wird dabei verzichtet, es gilt immer auf dem ganzen Netz derselbe Polynomgrad; lediglich an lokalen Störungen wie kleinen Teilflächenlasten wird automatisch ein höherer Polynomgrad als in der Umgebung gewählt. Typische Polynomgrade liegen bei Platten- und Scheibenberechnungen zwischen 5 und 8.

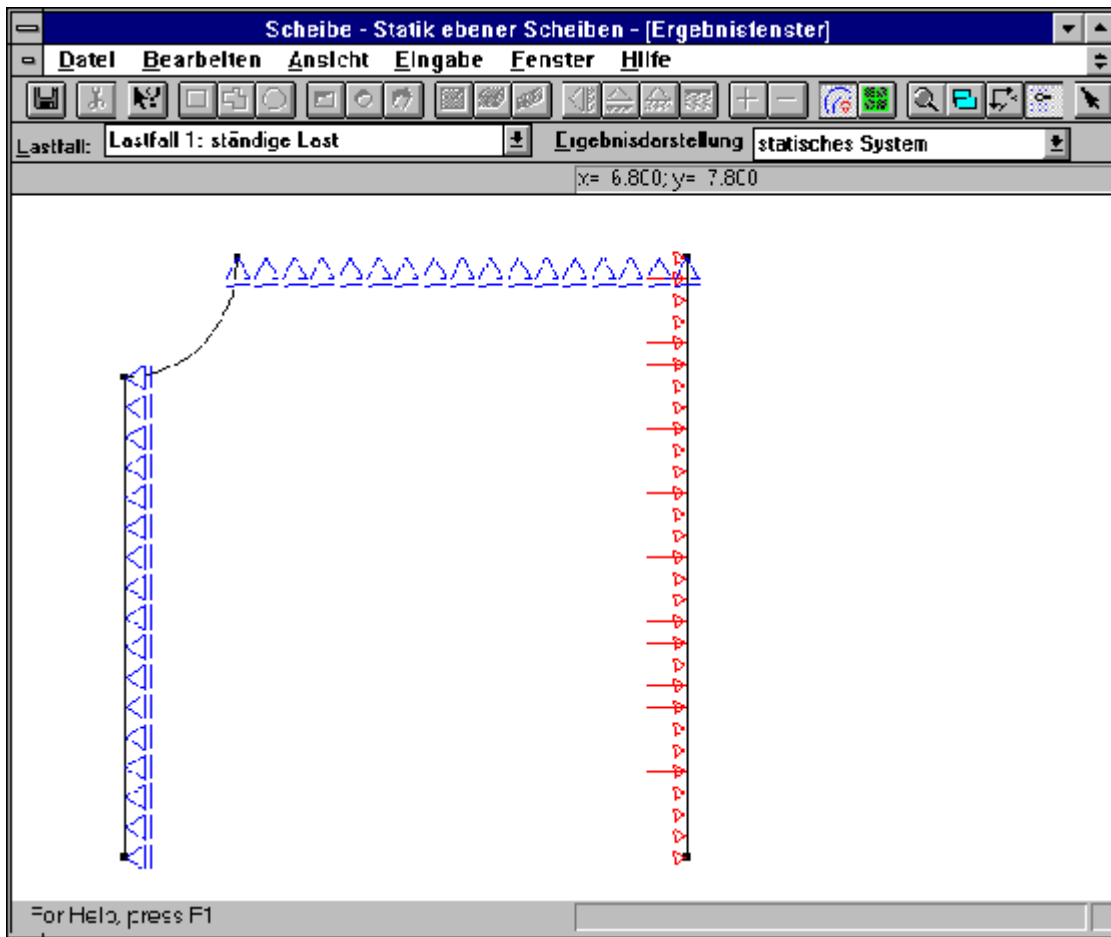




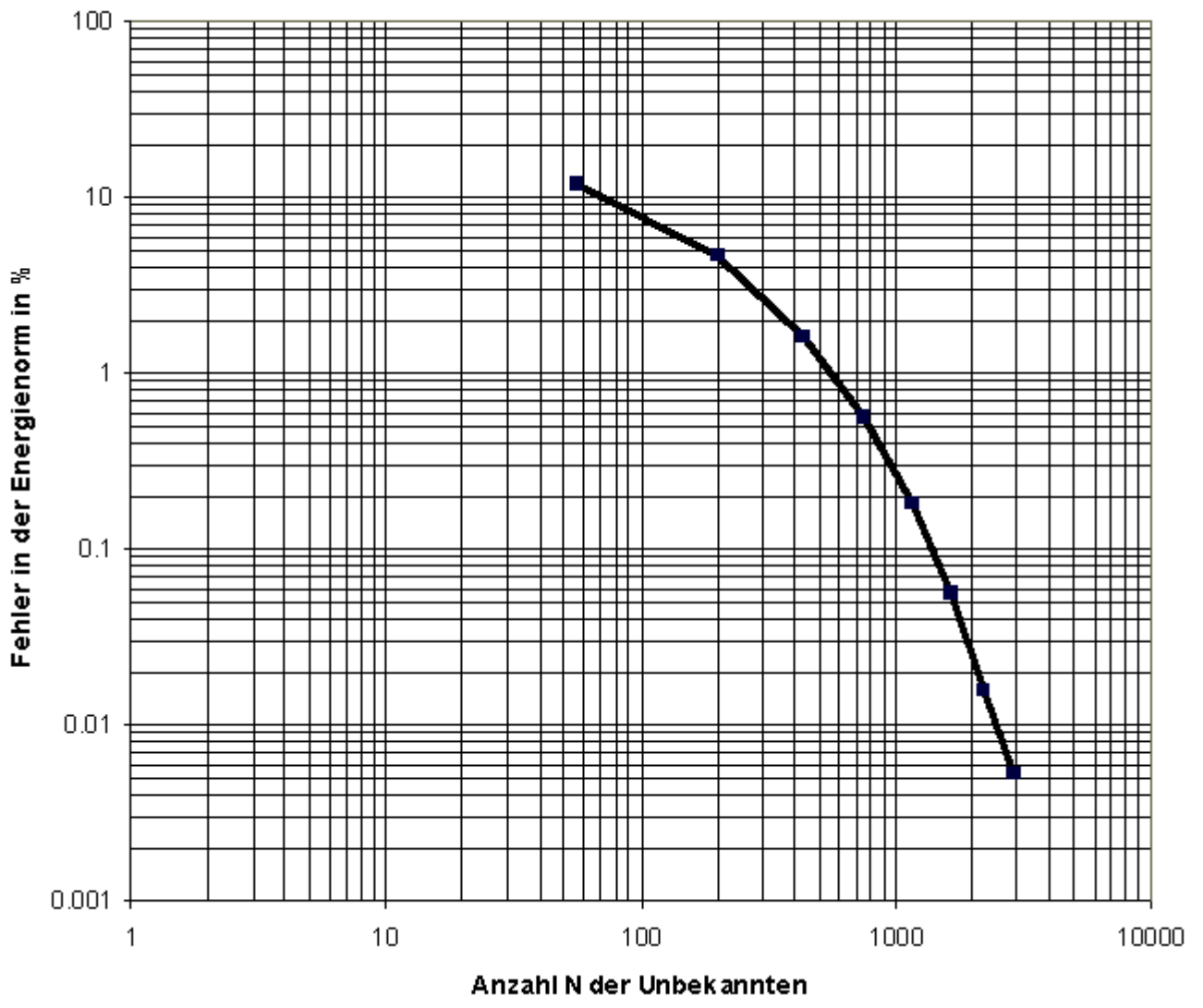
(Fig. 2) Typische Vernetzung einer Scheibe für die p -Version der FEM. - Typical mesh design for the p -version of the FEM.

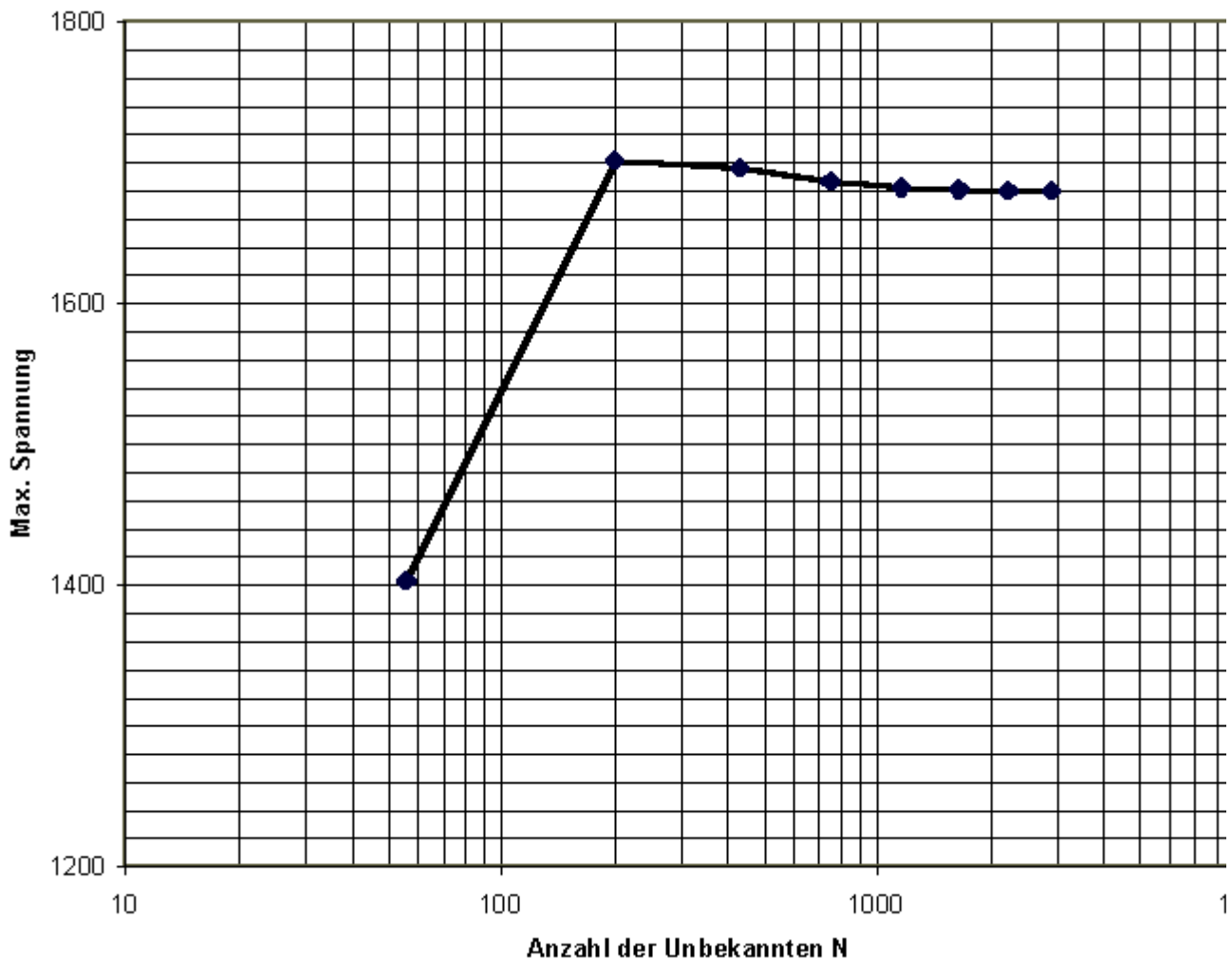
Idee dieses gegenüber der Adaptivität mittels lokaler Indikatoren vereinfachten Vorgehens ist es, den größten Teil der Vorzüge des adaptiven Verfahrens zu retten, den Berechnungsablauf jedoch möglichst einfach zu halten, damit das Verfahren effizient programmiert werden kann und die Rechenzeiten klein bleiben, so daß eine nahezu unmittelbare Rückkopplung zwischen graphischer Eingabe und Berechnungsergebnis möglich bleibt. Fig. 2 zeigt ein typisches, a priori an kritischen Stellen verfeinertes Finite-Elemente-Netz der p -Version.

Neben der globalen Genauigkeitskontrolle, die sich aus einer Beobachtung der Konvergenz der gesamten Verformungsenergie der Struktur während der Polynomgraderhöhung ableiten läßt, hat der Statiker auch die Möglichkeit, die Konvergenz einzelner, punktueller Ergebnisse in einer Graphik zu untersuchen. Für die Konvergenz lokaler Ergebnisse gibt es zwar keine mathematische Theorie, aber eine Konvergenzkurve z.B. für einen bemessungsrelevanten Spannungswert gibt allemal eine bedeutend bessere Information als nur ein einziger Ergebniswert, wie er normalerweise aus einer klassischen Finite-Elemente-Berechnung ermittelt wird. Erscheint die Genauigkeit noch unzureichend, wird einfach durch eine einzige Eingabe ein höherer Polynomgrad angefordert und das Ergebnis neu berechnet. Fig. 3 zeigt die Konvergenz der Berechnung einer kreisförmig gelochten Scheibe unter einachsigen Zug (Fig. 3a). Dargestellt ist sowohl die Konvergenz eines globalen Energiekriteriums (Fig. 3b) als auch der Maximalspannung am Lochrand (Fig. 3c) bei schrittweiser Erhöhung des Polynomgrades.



(a)





(Fig. 3)

- a Berechnung einer gelochten Scheibe unter einachsigem Zug mit der p -Version der FEM.
 - b Konvergenz des globalen Genauigkeitskriteriums
 - c Konvergenz der Maximalspannung am Lochrand bei schrittweiser Erhöhung des Polynomgrads. -
- a Analysis of a plate with a circular hole under uniaxial tension .
 b Convergence of the global error.
 c Convergence of the maximum tensile stress at the edge of the hole in p -version finite element analysis.

Die p -Version der Finite-Elemente-Methode ist für ein modernes Statiksystem deshalb so interessant, weil sie neben der besonders guten Konvergenz auch durch flexible geometrische Modellierung und Ergebnisauswertung ausgezeichnet ist: In den Programmen des Verfassers wird die Geometrie der Elemente - unabhängig von den Ansatzfunktionen - parametrisch beschrieben, wie dies z.B. auch in CAD-Programmen üblich ist. Hat ein Element beispielsweise eine kreisförmige Berandung, so wird bei der Berechnung der Steifigkeit in jedem Falle die exakte Kreisgeometrie verwendet, nicht eine polynomiale Näherung. Hierbei kommt die "Blending"-Technik zum Einsatz [7]. Somit sind nur wenige Elemente zur Erfassung einer komplexen Geometrie nötig.

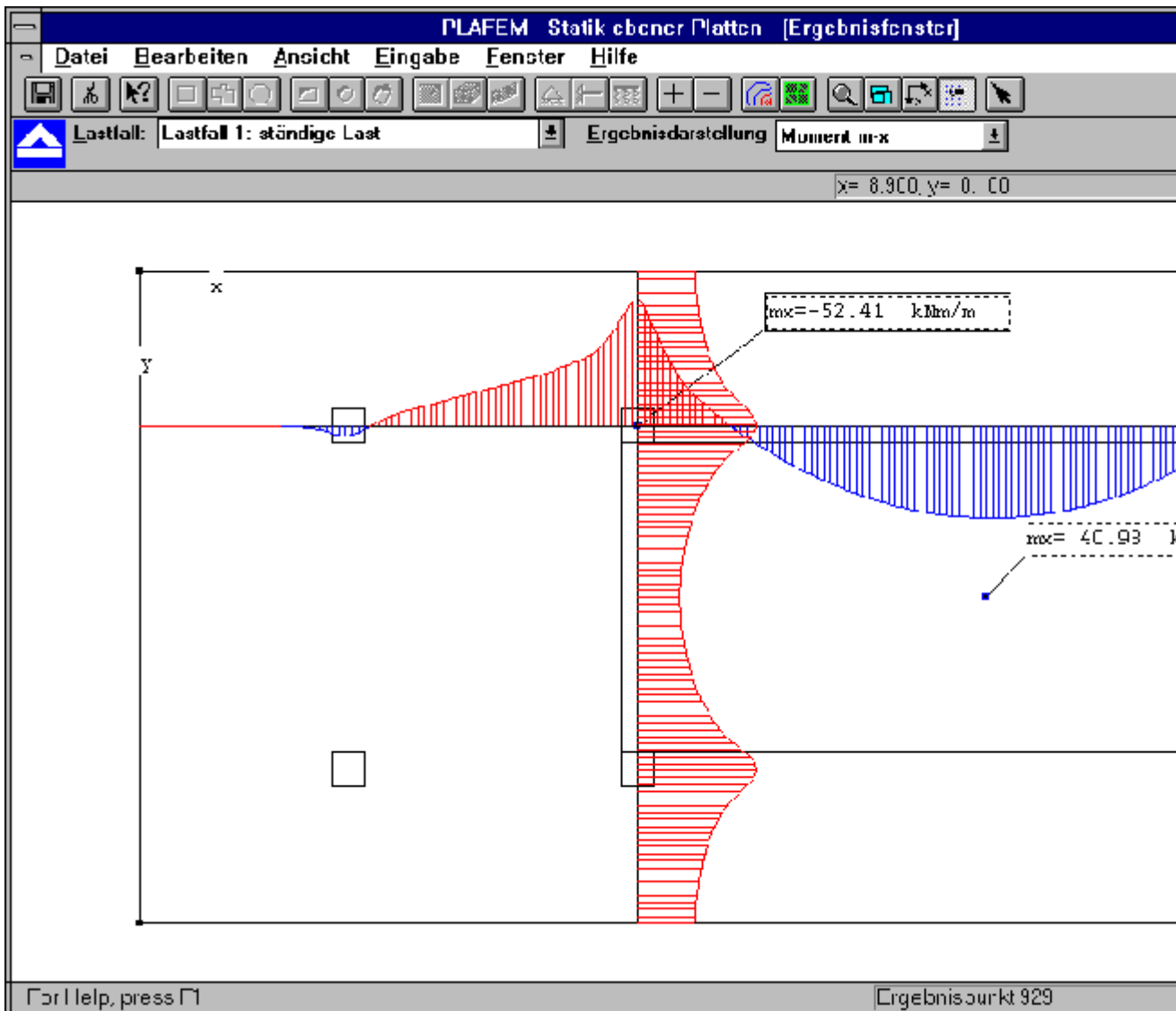
Versteifungseffekte bei Platten treten ab Polynomgrad 4 nicht mehr auf. Querkräfte können bei höheren Polynomgraden sehr gut aus den Gleichgewichtsbedingungen ermittelt werden [8]. Ergebnisse können zudem an jeder beliebigen Stelle der Elemente mit gleicher Zuverlässigkeit ermittelt werden, so daß man bei der Auswertung der Ergebnisse von der Geometrie des Finite-Elemente-Netzes weitgehend unabhängig ist.

Es sei nur am Rande erwähnt, daß die p -Version auch bei elastoplastischen Problemen durchaus konkurrenzfähig zur konventionelleren h -Version (Netzverfeinerung) ist, wie erste numerische Experimente zeigen [9].

Im Plattenprogramm des Verfassers wird weiterer Nutzen aus der p -Version der FEM gezogen, um das Problem der *Punktstützung* (punktgestützte Flachdecken, Pfahlgründungen usw.) in den Griff zu bekommen. Mit konventionellen FE-Programmen ist es fast unmöglich, punktgestützte Platten korrekt zu modellieren. Im hier vorgestellten System auf Basis der p -Version wird folgender Weg zur Lösung des Problems beschritten:

- Die Stütze wird durch einen elastisch gebetteten Teilbereich der Platte abgebildet. Realistische Bettungssteifigkeiten lassen sich aus der Stützengeometrie und dem statischen System der Stütze ableiten.
- Der kleine, elastisch gebettete Teilbereich, der die Stütze wiedergibt, muß nicht zur Finite-Elemente-Einteilung der Platte passen. Vielmehr wird eine ausreichende Genauigkeit durch Erhöhung des Ansatzgrades im Stützenbereich erreicht.
- Soll die Stütze gelenkig an die Platte angeschlossen sein, so entstehen bei dem Modell "elastisch gebetteter Teilbereich" bei unsymmetrischer Verformung unerwünschte Einspanneffekte. Um diese Einspanneffekte zu beseitigen, wird daher eine Nachiteration ausgeführt. Dabei wird angenommen, daß im Endzustand die Stütze eine gleichmäßige, konstante Pressung in die Platte einbringt und damit keinen Einspanneffekt hat (Man könnte auch jede andere symmetrische Pressungsverteilung ansetzen). Die reale Pressungsverteilung ist unsymmetrisch, und die Abweichungen von der gewünschten konstanten Verteilung werden zu einem "Restmoment" aufintegriert. Dieses Restmoment wird sukzessive durch Iteration mit konstanten Steifigkeiten "umverteilt", bis alle Stützen einspannungsfrei angeschlossen sind.

Fig. 4 zeigt die Ergebnisse (Biegemomente) einer "punktgestützten" Platte, die mit dieser Methode berechnet worden ist. Die Platte ruht auf sechs Stützen und ist im größeren rechten Feld durch eine konstante Flächenlast belastet. Fig. 4 zeigt die Biegemomente in zwei Schnitten durch die mittlere obere Stütze. Die Stützen tragen im Gleichgewichtszustand ihre Stützkraft durch eine konstante Pressung in die Platte ein. Entsprechend ist die Momentenverteilung automatisch so ausgerundet, wie dies der praktische Anwender für die Bemessung benötigt.



(Fig. 4) Berechnung einer punktgestützten Platte. Biegemomente m_x in zwei Schnitten durch die mittleren Stützen. - Analysis of a point-supported plate. Distribution of the bending moment m_x in two sections across one of the central pillars.

Auch andere bauspezifische Probleme wie z.B. Platten mit Unterzügen (oder Scheiben mit Randbalken) lassen sich mit Hilfe der p -Version der FEM besonders elegant beschreiben. Im Programm des Verfassers stehen dazu schubweiche Balkenelemente (Timoshenko-Balken) zur Verfügung, die kontinuierlich mit einem Plattenmodell verbunden werden können, an Stelle der üblichen "knotenweisen" Kopplung zwischen Euler-Bernoulli-Elementen und dem Plattenmodell.

Im Gegensatz zur Situation beim statischen System ist die Datenmenge, die ein Finite-Elemente-Modell (Netztopologie und Ansatzinformationen) beschreibt, groß. Die Menge an technischer Information aus einer Finite-Elemente-Berechnung, die gespeichert und weitergegeben werden muß, ist jedoch vom Umfang der Finite-Elemente-Datenmenge unabhängig und wird ausschließlich von den Erfordernissen der nachfolgenden Bemessung bestimmt. Für die Bemessung und Konstruktion der Baustruktur ist die Gestalt des Finite-Elemente-Modells somit völlig uninteressant. Lediglich die bemessungsrelevanten Ergebnisdaten, die aus einem Finite-Elemente-Modell gewonnen werden, sind von dauerhaftem Interesse. Finite-Elemente-Daten sind völlig ungeeignet zur direkten Benutzer-Programm-Interaktion und sollten daher im Hintergrund gehalten werden. Der Statiker besitzt in aller Regel nicht das Hintergrundwissen, um die Gestalt des Finite-Elemente-Netzes zuverlässig manuell verbessern zu können.

2.3 Ergebnismodell

Das fertig berechnete Finite-Elemente-Modell gestattet die Ermittlung konkreter Ergebniswerte an jeder beliebigen Stelle der modellierten Struktur.

Finite-Elemente-Modelle sind reine Hilfsmodelle, die nach Gebrauch sofort wieder gelöscht werden können, sobald alle signifikanten Ergebnisse berechnet sind. Dies wird spätestens dann deutlich, wenn man wirklich für jeden einzelnen Lastfall ein separates, individuell maßgeschneidertes Finite-Elemente-Modell entwickelt.

Als Träger der Ergebnisinformation sind die Finite-Elemente-Modelle einer Struktur damit aber ebenso ungeeignet, wie sie sich im vorausgehenden Abschnitt schon als ungeeignet erwiesen haben, als Träger der Information des statischen Systems zu fungieren.

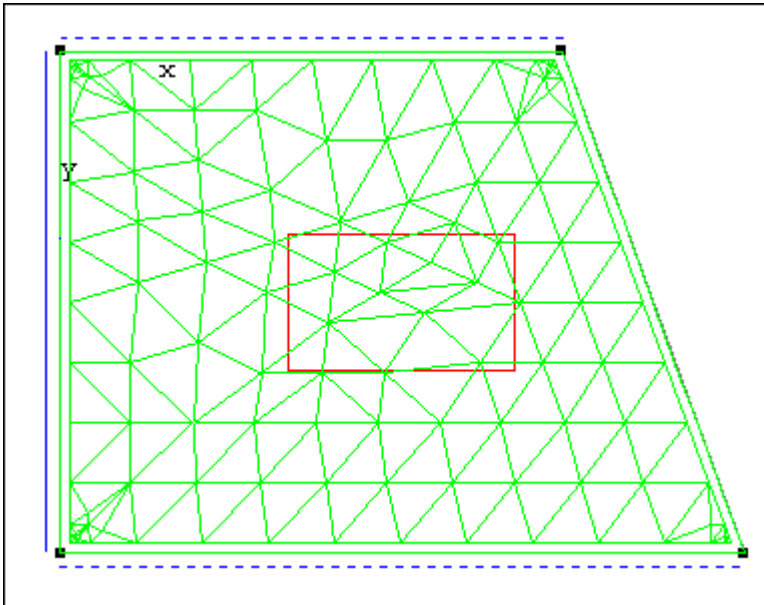
Man benötigt daher ein unabhängiges Ergebnismodell, das folgende Funktionen hat:

- Überlagerung der Lastfälle
- Schnelle Orientierung über den Ergebnisverlauf anhand von Höhenliniendarstellungen und beliebigen Schnitten.
- Identifikation von bemessungs- und konstruktionsentscheidenden Punkten, an denen die Ergebnisse auf jeden Fall ausgewertet werden müssen.
- Gezielte Auswahl von Ergebnissen für die Übernahme in andere Programme, z.B. in Bemessungsprogramme oder Nachweismodule.

Der Verfasser hat für die Erzeugung des Ergebnismodells in seinen eigenen Programmen folgenden Weg gewählt:

- Zur globalen Orientierung wird ein Dreiecks-Auswertungsnetz über die Struktur gelegt, dessen Feinheit der Benutzer je nach Bedürfnis beliebig regeln kann. An jedem Knoten dieses Netzes werden alle Ergebnisse berechnet. Dieses Netz dient dem Zeichnen von Höhenlinien oder der raschen, interaktiven Auswertung von Schnitten durch die Struktur zur ersten Orientierung.
- In einem zweiten Schritt, nach der globalen Orientierung anhand des Dreiecks-Auswertungsnetzes, legt der Benutzer seine endgültigen Auswertungspunkte fest. Die Finite-Elemente-Ergebnisse an diesen Stellen werden dann - zur Vermeidung von Interpolationsfehlern - noch nachträglich aus den Finite-Elemente-Ansätzen berechnet. Die endgültigen Auswertungspunkte können als Einzelpunkte oder als Punktfolge längs einer Schnittkurve gewählt werden.
- Alle Daten an den vom Benutzer definierten endgültigen Auswertungspunkten können anderen Anwendungsprogrammen zur Weiterverarbeitung zur Verfügung gestellt werden.
- An allen von Normen vorgeschriebenen Stellen werden automatisch Auswertungspunkte eingeführt; beim Stabwerksprogramm werden neben den Maxima und Minima der Schnittgrößenverläufe beispielsweise auch die Schnittgrößen an den Bemessungspunkten für die Schubbewehrung automatisch berechnet.

Fig. 5 zeigt ein typisches Ergebnismodell einer Plattenberechnung. Das Dreiecksnetz ist kein Finite-Elemente-Netz, sondern ein reines Auswertungsnetz, kann aber selbstverständlich mit dem Finite-Elemente-Freivernetzer generiert werden. Dieser Freivernetzer ist somit ein integraler Bestandteil des Baustatikprogrammes.



(Fig. 5) Typisches Auswertungsnetz bei einer Plattenberechnung - Typical result evaluation mesh in a plate analysis.

2.4 Graphische Interaktion zwischen Benutzer und Programm

Kaum ein kommerzielles Statikpaket kommt heute noch ohne eine graphische Benutzungsoberfläche aus, und die Attraktivität graphischer Ein- und Ausgabe ist mit Recht ungebrochen. Die Zeichnung ist die Sprache des Ingenieurs, und direktes graphisches Feedback trägt viel dazu bei, Fehler und Irrtümer zu vermeiden.

Dennoch sind auch bei der Gestaltung graphisch-interaktiver Programme im Bauwesen noch längst nicht alle Wünsche befriedigt. Prinzipiell kann die Benutzer-Programm-Interaktion einem der beiden folgenden Konzepte folgen:

1. Kommandoorientierte Interaktion: Diese entspricht der traditionellen textbasierten Kommunikation zwischen Benutzer und Programm mit Hilfe einer Kommandosprache (z.B. *"delete file"*). Übertragen auf die graphisch-interaktive Bedienung eines Programmes bedeutet dieses Vorgehen, daß der Benutzer aus einem Menü eine gewünschte Aktion und den Typ des Objektes auswählt, auf das die gewählte Aktion angewendet werden soll, etwa *"Löschen-Auflager"* oder *"Löschen-Last"*. Man hat dann nur noch ein konkretes Objekt mit einem Zeigegerät anzugeben, und die gewünschte Aktion wird ausgeführt.
2. Objektorientierte Interaktion: Zuerst selektiert der Benutzer eines oder mehrere Objekte, und erst danach wird eine Aktion ausgewählt, die auf die selektierten Objekte angewendet werden soll. Nach diesem Prinzip funktionieren inzwischen die meisten Standard-Büroprogramme. In vielen Fällen kann dabei auf die Auswahl eines Menüpunktes ganz verzichtet werden und die gewünschte Aktion direkt aus einer Benutzerhandlung geschlossen werden, z.B. Verschieben von Objekten durch Ziehen der Maus bei gedrückter Maustaste.

Entwickeln traditioneller Berechnungsprogramme liegt die erste Methode nahe, da sie völlig analog zu traditionellen Programmsteuerungen ist und keinen durchgreifenden Umbau eines existierenden Programmes, sondern lediglich die "Visualisierung" der bisherigen Kommandosprache erfordert. Dementsprechend häufig sind solche Benutzungsoberflächen zu finden. Sie haben jedoch folgende Nachteile:

- Da ein Statikprogramm eine Vielzahl möglicher Aktionen kennt und ebenfalls eine Vielzahl möglicher Objekttypen, auf die diese Aktionen angewendet werden können, kennt, entstehen

sehr schnell sehr große und unübersichtliche Menüstrukturen, da sich die Anzahl der Aktionen mit der der Objekte multipliziert.

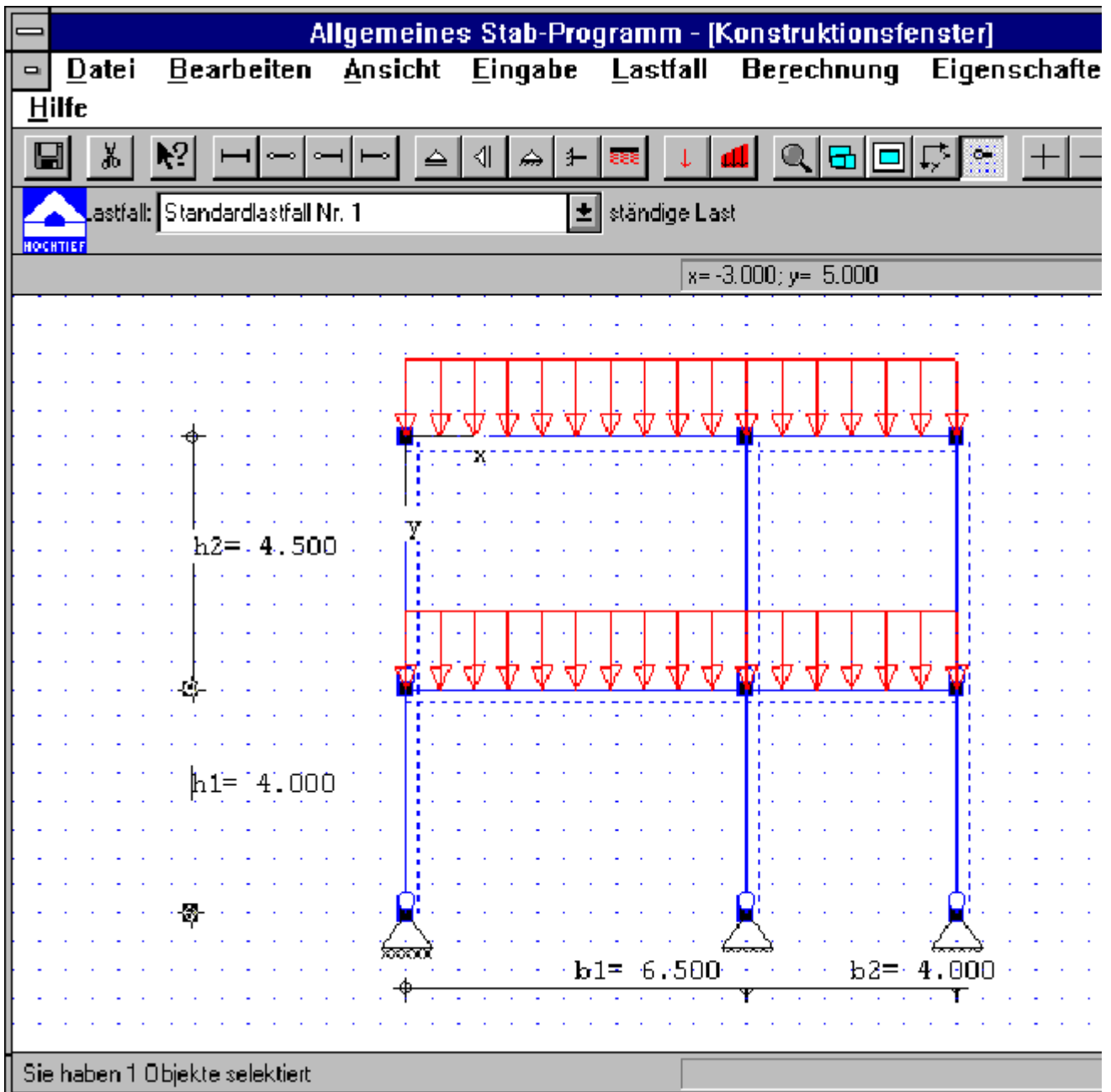
- Benutzer eines graphisch-interaktiven Programmes versuchen intuitiv, Objekte direkt zu ergreifen. Dies ist jedoch in der 1. Methode nicht als primärer Schritt vorgesehen. Ein Programm, das graphische Manipulation von Objekten anbietet, aber zuerst die Auswahl einer Aktion fordert, wird als umständlich und "unnatürlich" empfunden. Benutzer, die eine im Hintergrund stehende, lediglich "visualisierte" Kommandosprache nicht kennen, kommen mit einem solchen Statikprogramm nicht zurecht.

Will man die objektorientierte Interaktion implementieren, so bleibt fast nur die Programmierung in einer objektorientierten Programmiersprache wie C++. Dank der Hierarchie der Klassen kann man in einer solchen Programmiersprache für alle Objekttypen, die z.B. durch graphische Interaktion verschoben oder gelöscht werden können, das gewünschte Protokoll in einer Oberklasse festschreiben und dann für jede spezielle Klasse in überladenen Funktionen entsprechend implementieren.

Grundfunktionen graphisch manipulierbarer Objekte sind *Selektieren*, *Deselektieren*, *Verschieben*, *Löschen*, *Kopieren*, *Eigenschaften bearbeiten*, *Verzerren*. Erfüllen alle Objektklassen graphisch darstellbarer Objekttypen eines Programmes ein solches Protokoll, so kann man auch ganze Gruppen von Objekten verschiedenen Typs gemeinsam manipulieren, was mit der kommandoorientierten Methode nicht möglich ist.

Eine weitere Grundvoraussetzung für maximalen Nutzeffekt einer graphisch-interaktiven Programmbedienung ist das unmittelbare Feedback. Im Idealfall sollen sofort alle Auswirkungen einer Aktion sichtbar werden. Diese Anforderung schließt Programmsysteme, die der traditionellen Aufteilung in Präprozessor, Rechenzeit und Postprozessor folgen, aus. Das interaktive Statik-Programm der Zukunft vereinigt diese drei Komponenten in einem einzigen Programm.

Ein Programm, das zur graphisch-interaktiven Konstruktion eines statischen Systems dient, hat viel Ähnlichkeit mit einem CAD-Programm, und Techniken, die im CAD die Arbeit vereinfachen, sind auf das Statik-Programm übertragbar. Besonders leistungsfähig erscheint dem Verfasser das Konzept der *parametrischen Konstruktion*. Parametrische Konstruktion gestattet dem Benutzer, die Geometrie und auch alle anderen Eingabedaten eines Problems mit Hilfe von formelmäßigen Zusammenhängen anhand einiger weniger Parameter zu definieren, sich also statische Modelle zu schaffen, die abgespeichert werden können und beim nächsten Einsatz einfach durch Aktualisieren der Parameter an eine neue Variante der Konstruktion angepaßt werden können. Beispielsweise können alle Geometrie-, Last- und Lastfallüberlagerungsdaten eines Kehl balkendachstuhls durch ganz wenige Parameter wie Dachneigung, Spannweite, Eigengewicht der Konstruktion usw. beschrieben werden. Der Benutzer eines Statik-Programmes, das die parametrische Konstruktion vorsieht, kann also sein eigenes, maßgeschneidertes Kehl balkendachprogramm erstellen, indem er einfach einmal die Struktur eingibt, parametrisiert und abspeichert. Der Einsatz einer Vielzahl von Einzelprogrammen etwa für Durchlaufträger, Rahmen, Stockwerkrahmen, Dachkonstruktionen usw. wird durch ein eigenes, einheitliches Werkzeug ersetzt, daß sich zudem beliebig den eigenen Wünschen anpassen läßt. Außerdem kann man im Fall eines nicht ganz "standardmäßigen" Tragwerks, z.B. eines unsymmetrischen Daches, immer noch die parametrisierte Konstruktion durch Ad-Hoc-Eingaben an die aktuellen Gegebenheiten anpassen, wozu das starre Spezialprogramm keine Chance gibt. Fig. 6 zeigt ein Beispiel einer parametrischen Konstruktion im Stabwerksprogramm des Verfassers. Die Abmessungen können einfach durch Anklicken der Maßlinien und Eintragen des neuen Wertes angepaßt werden. Die gesamte Struktur wird dann konsistent aktualisiert.



(Fig. 6) Arbeiten mit parametrisierten Vorlagen: Ein Stockwerksrahmen wird durch vier Parameter beschrieben. - Working with parametrized templates: Multi-storeyed frame defined by four parameters.

2.5 Vereinheitlichung von Programmen durch Orientierung an industriellen Quasistandards

Marktbeherrschende industrielle Produkte besitzen normsetzende Kraft. Im Technischen Büro werden fast ausschließlich PCs mit Intel-Prozessor und mit der einen oder anderen Variante des Betriebssystem Windows von Microsoft eingesetzt. Die Produkte, die von Microsoft für den breiten Markt der allgemeinen Bürosoftware (Textverarbeitung, Tabellenkalkulation, relationale Datenbanken) angeboten werden, folgen heute detailliert gefaßten, einheitlichen Gestaltungsrichtlinien.

Demgegenüber zeichnet sich die derzeit verfügbare kommerzielle Bausoftware, soweit sie über graphische Eingabeoberflächen verfügt, noch durch ein buntes Vielerlei unterschiedlichster Oberflächengestaltungen aus. Dies ist einer der Gründe dafür, daß sich Benutzer beim Einstieg in eines der bauspezifischen Programme oft so schwer tun, während bei allgemeiner Bürosoftware selbst

Mitarbeiter, die kaum Hintergrundwissen zu Computern besitzen, sehr schnell mit den Programmen zurechtkommen.

Einheitliches Aussehen und einheitliche Funktionalität von Programmen steigern aber nicht nur die Effizienz der Arbeit, sondern helfen auch, Interpretations- und Bedienungsfehler zu vermeiden und dienen damit der Qualitätssicherung der computergestützten Arbeit.

Es ist also nicht damit getan, die eigene, bisher beispielsweise auf DOS mit einer eigenen Oberfläche laufende Applikation unter Beibehaltung des alten "Look and Feel" einfach auf Windows zu übertragen und dies dann als Windows-Version des alten Programmes zu verkaufen; vielmehr ist ein komplettes Redesign notwendig. Selbst die Firma AutoDesk, die selbst mit dem DXF-Format einst einen Industriestandard gesetzt hat, hat die neue Version 13 des bekannten AutoCAD-Programmes nahtlos in die Windows-Oberfläche eingebunden. Eine vergleichbare Entwicklung sollten auch die praktischen Nutzer von Bausoftware von den einschlägigen Systemhäusern fordern.

Viele Menüpunkte, die in allgemeiner Bürosoftware notwendig sind, sind auch in einem Statikprogramm sinnvoll, und sollten dort an derselben Stelle mit derselben Funktionalität zu finden sein. Auch Maus- und Tastaturfunktionalität sollten sich so eng als möglich an die Vorgaben durch Standardbüroprogramme anlehnen.

Nebenprodukt einer Angleichung an einen Industriestandard "Windows" ist die fast kostenlos Integration der entstehenden Produkte auf einem Niveau, das mindestens mit den meisten bisherigen Integrationsbemühungen über genormte Schnittstellen mithalten kann. Stichworte wie Kopieren von Texten oder Graphiken über die Zwischenablage, Einbinden kleinerer Programme in klassische Bürosoftware durch "Object Linking and Embedding" und Anbindung von spezieller Software an Standardprogramme durch Makroprogrammierung mögen genügen.

So entsteht eine Statik-Umgebung auf dem PC, die die Bearbeitung des Endproduktes des Statikers - des Statischen Berichtes - mit professionellen Werkzeugen und in einer Bedienungsqualität gestattet, die durch spezielle "Statik-Editoren" nie erreicht wird. An die Stelle eines hierarchischen Mitwirkens von Komponenten unter dem Dach eines "Statik-Editors" tritt ein gleichberechtigtes Zusammenwirken heterogener Spezialprogramme, die für ihren jeweiligen Anwendungszweck maßgeschneidert sind.

3 Schlußbemerkung

Folgt man den Leitgedanken des vorliegenden Aufsatzes, so kommt man zu einer computerbasierten Arbeitsweise, die erstaunliche Ähnlichkeit mit der althergebrachten Handrechnung mit Tabellenwerk, Papier und Bleistift hat. Der entscheidende Fortschritt besteht darin, daß ein solches modernes Statikprogramm ungleich flexibler und leistungsfähiger als die konventionelle Tabelle ist, da es jede beliebige Geometrie, Belastung, Lastfallüberlagerung usw. gestattet und zugleich optimal für das Zusammenwirken mit Standard-Bürosoftware vorbereitet ist, so daß die gesamte Erstellung des Statischen Berichtes nahtlos auf dem PC abgewickelt werden kann.

Der Statiker konzentriert sich in einem solchen System wieder auf die korrekte Bildung eines adäquaten statischen Modells, die Auswertung einiger weniger charakteristischer Ergebnisse, deren manuelle Plausibilitätskontrolle und Beurteilung und eine darauf aufbauende Bemessung und Konstruktion. Alle anderen Informationen bleiben im Hintergrund, insbesondere alle Finite-Elemente-Informationen. Daß die numerische Berechnung mit finiten Elementen erfolgt ist, spielt in einem solchen System nur noch eine untergeordnete Rolle.

Will man die Konzentration des Statikers zurück auf das statische Modell und die Ergebnisauswertung lenken, so sind die Neuentwicklungen der Finite-Elemente-Theorie - vollautomatische Freivernetzung, automatische Diskretisierung aller Lasten und Randbedingungen, Genauigkeitsabschätzung und

Adaptivität - integrale Bestandteile eines ingenieurgerechten Modellierungs- und Berechnungsprogrammes.

Literatur

1. **Beucke, K.:** Stand der Integration von Statik und Bemessung im Entwurfs- und Konstruktionsprozeß. Bauingenieur 71 (1996), 67-72.
2. **Stein, E.:** Anspruch und Wirklichkeit in der Bausoftware. Bauingenieur 71 (1996), A9.
3. **Stein, E.; Ohnimus, S.:** Dimensional Adaptivity in Linear Elasticity with hierarchical test spaces for h- and p-refinement processes. Eng. Comp. 12 (1996), 107-119.
4. **Szabó, B. A.; Sahrman, G.:** Hierarchic plate and shell models based on p-extension, Int. J. Numer. Meth. Engrg. 26 (1988), 1855-1881.
5. **Szabó, B. A.; Babuska, I.:** Finite Element Analysis. John Wiley & Sons, New York (1991).
6. **Rank, E.; Schweingruber, M.; Sommer, M.:** Adaptive mesh generation and transformation of triangular to quadrilateral meshes. Comm. Appl. Numer. Meth. Engrg. 9 (1993), 121-129.
7. **Holzer, S.; Haüßer, C.:** Ergebnisverbesserung von Finite-Element-Berechnungen durch exakte Geometriemodellierung. Forschung im Ingenieurwesen 56 (1990), 65-70.
8. **Rank, E.:** mündliche Mitteilung (1996).
9. **Holzer, S.; Yosibash, Z.:** The p -version of the finite element method in incremental elasto-plastic analysis. Int. J. Numer. Meth. Engrg. 39 (1996), 1859-1878.

S. M. Holzer

Universität Stuttgart, Informationsverarbeitung im Konstruktiven Ingenieurbau, Pfaffenwaldring 7, D-70550 Stuttgart

Dr.-Ing. Stefan M. Holzer war bis September 1995 Mitarbeiter der HOCHTIEF AG in Frankfurt und befaßte sich dort unter anderem mit der Entwicklung neuer Statik-Software. Seither ist er Professor an der Universität Stuttgart und leitet das dort neu eingerichtete Fachgebiet Informationsverarbeitung im Konstruktiven Ingenieurbau, das an der Universität Stuttgart das Lehrgebiet Bauinformatik vertritt.

Der vorliegende Beitrag ist Herrn Prof. Dr.-Ing. Heinrich Werner, München, aus Anlaß seines 65. Geburtstages gewidmet. Der Verfasser weiß sich Herrn Prof. Werner, der zu den Gründungsvätern des Faches Bauinformatik zählt, in vielfacher Weise für Anregungen und Unterstützung zu Dank verpflichtet und glaubt, daß der vorliegende Beitrag in direkter Nachfolge der langjährigen Tätigkeit von Herrn Prof. Werner in München steht.



[Home] [[Aktuelle Veröffentlichungen](#)]

01.08.1996