

Zeitschrift: Schweizer Ingenieur und Architekt
Herausgeber: Verlags-AG der akademischen technischen Vereine
Band: 101 (1983)
Heft: 6: Prof. Dr. Bruno Thürlimann zum 60. Geburtstag

Artikel: Flowers: ein neues Finite-Element-Programm für Lehre, Forschung und Praxis
Autor: Anderheggen, Edoardo
DOI: <https://doi.org/10.5169/seals-75053>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 24.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

FLOWERS: Ein neues Finite-Element-Programm für Lehre, Forschung und Praxis

Von Edoardo Anderheggen, Zürich

Am Institut für Informatik der ETH Zürich wird seit über drei Jahren an der Entwicklung des neuen, für lineare und nichtlineare statische und dynamische Berechnungen anwendbaren Computerprogramms FLOWERS gearbeitet. Die dabei verfolgten Ziele sowie die wichtigsten Eigenschaften des neuen Programms werden erörtert.

A l'Institut d'informatique de l'EPF Zurich on travaille depuis plus de trois ans au développement du nouveau programme d'ordinateur FLOWERS capable de résoudre une vaste gamme de problèmes linéaires et nonlinéaires statiques et dynamiques par la méthode des éléments finis. Les buts de ce projet ainsi que les caractéristiques principales du nouveau programme sont discutées.

At the Institute of Informatics of the Swiss Federal Institute of Technology in Zurich the new computer program FLOWERS capable on solving linear and non-linear static and dynamic structural analysis problems is being developed since over three years. The aims of the project as well as some important properties of the new program are discussed.

Vorgeschichte und Ziele

Gegen Mitte der sechziger Jahre wurde am Institut für Baustatik und Massivbau der ETH Zürich unter der Leitung von Prof. Bruno Thürlimann das aus dem Massachusetts Institute of Technology stammende Computerprogramm STRESS [1] überarbeitet und den Bauingenieurstudenten sowie der Praxis zur Verfügung gestellt. Historisch gesehen hat das zur Durchführung linearer statischer Stabtragwerksberechnungen während über zehn Jahren viel verwendete Programm STRESS eine wichtige Rolle gespielt. Es stellte nämlich für viele schweizerische Bauingenieure den allerersten Kontakt mit Computern dar. Bemerkenswert ist auch die problemorientierte, mit einem beträchtlichen Programmieraufwand verbundene Eingabesprache des Programms STRESS, die seine Benützung wesentlich vereinfacht. Man verstand nämlich zum ersten Mal, wie wichtig es sei, bei derartigen praxisorientierten baustatischen Programmen nicht nur die gewünschten Berechnungen durchzuführen, sondern auch die Programmbenützer möglichst zu unterstützen. Dies war damals nicht so selbstverständlich, wenn man bedenkt, dass das Know-how und der Aufwand für die Entwicklung benutzerfreundlicher Computerprogramme um ein Vielfaches grösser sein kann als für solche, die an sich die genau gleichen technischen Probleme lösen, jedoch ohne Rücksicht auf Benutzerkomfort. Dafür ist inzwischen sogar die neue Berufsgattung der professionellen Informatiker entstanden.

Nach STRESS kamen verschiedene Programme zur Berechnung von Flä-

chentragwerken nach der Methode der finiten Elemente (FE), wovon FLASH [2] heute in der Praxis am meisten verwendet wird.

Erwähnenswert ist aber hier vor allem das gegen Mitte der siebziger Jahre entwickelte Programm STATIK [3]. Es richtet sich an die gleiche Art von Benützern wie STRESS und hat auch im wesentlichen den gleichen, an sich sehr beschränkten Anwendungsbereich (linear-elastische statische Stabtragwerksberechnungen). STATIK, gestützt auf die inzwischen gesammelten Erfahrungen, bietet jedoch seinen Benützern wesentlich mehr Komfort als STRESS. Seine Eingabesprache ist nicht nur formatfrei und problemorientiert, sondern auch mit Hilfe äusserst kompakter und übersichtlicher Syntaxdiagramme beschreibbar (mehr darüber im Zusammenhang mit dem Programm FLOWERS). Es kann zudem verschiedene Resultate in graphischer Form ausgeben, was für baustatische Anwendungen eine unbestrittene Notwendigkeit darstellt. Schliesslich erlaubt STATIK die selektive Änderung von Problemdateien und die Wiederholung bestimmter Berechnungsschritte auf Grund der früher gespeicherten bzw. geänderten Problemdateien. Nach einem Fortbildungskurs an der ETHZ im Herbst 1977 fand STATIK Eingang in die Praxis, wo es heute in der Schweiz und neuerdings vermehrt auch in der Bundesrepublik Deutschland und Österreich zum alltäglichen Werkzeug vieler Bauingenieure geworden ist. STATIK hat sich zudem als geeignetes Instrument der Lehre an der Abteilung für Bauingenieurwesen der ETHZ erwiesen, wo es innerhalb verschiedener Vorlesungen der Vertie-

In diesem Heft veröffentlichen wir den ersten Teil einer Reihe von Artikeln, die zum 60. Geburtstag von Prof. Dr. Bruno Thürlimann geschrieben worden sind. Der abschliessende Teil folgt in Heft 7 vom 10. Februar.

fungsrichtung Konstruktion regelmässig verwendet wird.

Der Anwendungsbereich des Programms STATIK wurde gezielt auf Berechnungen beschränkt, die der durchschnittliche konstruktiv tätige Bauingenieur täglich braucht und in der Regel ohne Schwierigkeiten überblickt. Im besonderen geht es dabei um Probleme, die keine numerische Approximation ausser die der allgemein bekannten einfachen Stabtheorie benötigen. Die bei mehrdimensionalen oder bei dynamischen oder bei nichtlinear-elastischen FE-Berechnungen oft heikle Frage nach der Wahl eines genügend genauen, jedoch nicht unnötig aufwendigen mathematischen Modells stellt sich deswegen nicht. Dies vereinfacht die Anwendung des STATIK-Programms sehr und stellt sicherlich einen wichtigen Grund seiner heutigen Verbreitung dar. Andererseits haben die erwähnten Anwendungsbeschränkungen zur Folge, dass andere Bedürfnisse vom Programm STATIK nicht erfüllt werden können.

Zuerst sind die Bedürfnisse der Lehre bezüglich FE-Methoden in den höheren Semestern der Abteilung für Bau-, für Maschineningenieurwesen und möglicherweise für Informatik sowie für Nachdiplom- und Fortbildungskurse zu erwähnen, wo es heute sicherlich nicht nur um linear-elastische statische Stabprobleme gehen kann. Dabei müssen nämlich die Studenten selbständige Erfahrungen bei der Verwendung von FE-Programmen unbedingt sammeln können, wofür im Haus entwickelte, vom Dozenten und Assistenten in ihrer internen Arbeitsweise klar überblickbare Computerprogramme wenn möglich verwendet werden sollen. Kommerziell verfügbare FE-Programme sind nicht nur oft sehr schwer überblickbar, sondern auch sonst für Lehrzwecke ungeeignet, weil sie dafür gar nicht geplant wurden.

Es sind dann die Bedürfnisse der Hochschulforschung zu erwähnen. Wenn ein Forscher eine besondere Art von Problemen nach der Methode der finiten Elemente, jedoch mit z.T. eigenen Prozeduren lösen will (z.B. wenn es um die Erfassung besonderer Materialeigenschaften mit Hilfe eines speziellen mathematischen Modells geht), ist er mit einer Reihe von Fragen konfrontiert, die sich bei jeder FE-Anwendung im-

mer wieder stellen, die aber in keinem direkten Zusammenhang mit seiner spezifischen Problemstellung stehen und die eine grosse Belastung darstellen können. Wie später noch erklärt wird, ist es jedoch in vielen Fällen möglich, in einem bestehenden, dafür speziell konzipierten System durch nur geringfügige Programmanpassungen eigene Prozeduren einzubauen und auszutesten. Kommerzielle FE-Programme sind dafür in der Regel ungeeignet, nicht zuletzt weil die Programmentwickler, auf deren Unterstützung man bei Programmänderungen meistens angewiesen ist, oft unerreichbar sind.

Schliesslich sind die Bedürfnisse der Praxis zu erwähnen, wo FE-Programme im Bauingenieurwesen und in der Maschinenindustrie heute sehr viel verwendet werden. Dabei ist zu bemerken, dass, während man auf dem Hardwaresektor in den letzten Jahren stürmische Entwicklungen erlebt hat, die meistens heute viel verwendeten FE-Grossprogramme (NASTRAN, ASKA, MARC usw.) vor 10 bis 15 Jahren konzipiert und danach immer nur verbessert und geflickt wurden. Dahinter steckt nämlich eine derart grosse *Software-Entwicklungsarbeit*, dass es heute sehr schwierig für eine Privatfirma wäre, diese Programme durch völlig neukonzipierte zu ersetzen. Dies ist jedoch an einer Universität möglich und, unserer Ansicht nach, auch für die Praxis lohnend, weil vor allem bezüglich Benutzerkomfort und speziell bei nichtlinearen Anwendungen Verbesserungen zweifellos noch möglich sind.

Als Folge solcher Überlegungen entschloss man sich gegen Ende der siebziger Jahre, das neue Computerprogramm FLOWERS zu entwickeln, wofür bis heute schon etwa zehn Mann-Jahre Arbeit aufgewendet wurden. Dabei hat man versucht, ganz im Gegenteil zu STATIK, den Anwendungsbeereich so wenig wie möglich a priori zu begrenzen. FLOWERS kann im Prinzip *alle Arten von Tragwerken* sowie auch *nichtfestigkeitstheoretische Probleme* (Wärmeleitung, elektrische und magnetische Felder usw.) lösen. Es kann statische bzw. stationäre und dynamische bzw. instationäre, lineare und nichtlineare Berechnungen durchführen. Man hat zudem versucht, durch saubere Programmstrukturierung und durch klare Festlegung bestimmter Schnittstellen damit auch ein Instrument der *Hochschulforschung* zu schaffen. Fragen der *leichten Lernbarkeit* und der sonstigen *Benutzerfreundlichkeit* hat man im Blick auf die Bedürfnisse der Lehre und der Praxis viel Aufmerksamkeit geschenkt (wobei jedoch zu bemerken ist, dass die Benützung

des Programms FLOWRES, wegen seines breiten Anwendungsbereichs, im Gegensatz zu STATIK, nicht jedermanns Sache ist). Völlig *neue Wege* ist man bezüglich *nichtlinearer Anwendungen* gegangen, wie in einem in dieser selben Zeitschrift vor kurzem erschienenen Artikel [4] im Detail berichtet wurde. Schliesslich ist zu erwähnen, dass FLOWERS ein in Standard-FORTRAN geschriebenes Programm ist, das heute sowohl auf der DEC-10-Anlage des Zentrums für Interaktives Rechnen (ZIR) als auch auf den CDC-CYBER-Anlagen des Rechenzentrums der ETH Zürich läuft und das leicht auf jedem anderen Gross- bis Minicomputer installiert werden könnte.

Programmstruktur

Eine FE-Berechnung mit dem Programm FLOWERS setzt sich aus *zwei Arten von Verarbeitungsschritten* zusammen, die sich klar voneinander unterscheiden lassen, nämlich aus solchen, die das *globale System* und solchen, die *jedes einzelne Element* betreffen. In der hiernach verwendeten Terminologie werden die ersten von den «Systemroutinen», die zweiten von den «Elementroutinen» durchgeführt. Aufgaben der Systemroutinen sind z.B. das Lesen der Systemdaten, die Aufstellung und Lösung der globalen Gleichungssysteme oder die Datenverwaltung auf Primär- und Sekundärspeicher. Aufgabe der Elementroutinen ist die Bereitstellung der numerischen Koeffizienten, die das Verhalten jedes einzelnen Elementes numerisch beschreiben und die dann von den Systemroutinen weiterverarbeitet werden. Die Systemroutinen hängen vom Lösungsverfahren, nicht vom Problemtyp ab. So verlangen beispielsweise statische und dynamische Berechnungen auf globaler Systemebene eine völlig verschiedene Behandlung und damit andere Systemroutinen. Diese brauchen aber z.B. zwischen einer Rahmen- und einer Schalenberechnung nicht zu unterscheiden, da sie die lokalen Elementkoeffizienten, die sie einmal von Rahmenelementroutinen, einmal von Schalenelementroutinen übermittelt erhalten, nach identischen Prozeduren weiterverarbeiten. Dies macht es möglich, FE-Programme mit einer aus vielen Elementroutinen bestehenden und beliebig erweiterbaren Elementbibliothek zu schreiben, die völlig verschiedenartige Probleme behandeln können.

Die meisten grossen FE-Programme arbeiten nach diesem Prinzip, so auch das Programm FLOWERS, bei dem die

scharfe Trennung zwischen Systemroutinen und Elementroutinen auch deswegen wichtig ist, weil FLOWERS ein *Forschungsinstrument* werden soll. Dies bedingt nämlich, dass neue Elementroutinen für besondere Problemarten, Materialeigenschaften, Lasteneinwirkungen usw. von fremden Programmierern ohne Detailkenntnisse über die Arbeitsweise der Systemroutinen leicht ausprobiert werden können.

FLOWERS besteht aus *acht unabhängigen Programmen*, sogenannten Modulen, die verschiedene Aufgaben übernehmen. Einige dieser Module bestehen nur aus Systemroutinen und sind folglich völlig anwendungsunabhängig (z.B. das Modul SOLVE, das nicht zu wissen braucht, ob die zu lösenden Gleichungssysteme beispielsweise aus Schalen- oder aus Wärmeleitungsrechnungen stammen). Andere, anwendungsunabhängige Module bestehen aus System- und Elementroutinen (z.B. SYSIN). Davon existieren je nach Anwendungsgebiet verschiedene Versionen, die aus den gleichen Systemroutinen, jedoch aus verschiedenen Elementroutinen zusammengesetzt sind. So enthalten beispielsweise die für räumliche Stab- und Schalenberechnungen verwendeten anwendungsabhängigen Module räumliche Rahmen-, Fachwerk-, Schalen- und Membranelemente, die die gleiche Art von Knotenparameter haben und folglich miteinander kompatibel sind, nicht aber z.B. Platten- oder Wärmeleitungselemente.

Bild 1 stellt schematisch die Module des Programms FLOWERS dar und gibt an, welche davon neben Systemroutinen auch noch Elementroutinen enthalten. Das Bild zeigt zudem den sich über externe Dateien abspielenden Datenfluss zwischen Modulen und damit die mögliche Reihenfolge der Modulausführungen. Um die auf globaler Systemebene vorhandenen Möglichkeiten des Programms FLOWERS zu zeigen, sollen im folgenden noch die *Aufgaben jedes Moduls* kurz erörtert werden (über eigentliche Anwendungsbereiche wird unten im Zusammenhang mit der Elementbibliothek eingegangen):

SYSIN wird für ein bestimmtes Problem zuerst ausgeführt und dient in erster Linie zur Spezifikation sowohl der globalen System- und Lastdaten (Knotenkoordinaten, Elementinzidenzen, Lastfaktoren usw.) als auch der elementspezifischen Daten (z.B. Materialbeiwerte). Elementmaschen im ein-, zwei- und dreidimensionalen Koordinatenraum, bestehend aus 1- bis 20 knötigen Elementen mit 1 bis 12 Parametern je Knoten, können spezifiziert werden. Die an SYSIN angeschlossenen Elementroutinen interpretieren die ele-

mentspezifischen Eingabedaten und bestimmen die später für lineare und möglicherweise nichtlineare Berechnungen notwendigen Elementdaten (bei linearen Elementen: elastische und geometrische Steifigkeits-, Lasten-, Massen- und Spannungsmatrizen). Diese, zusammen mit den globalen System- und Lastdaten, werden auf Sekundärspeicher geschrieben.

SOLVE dient zur Lösung linearer Gleichungssysteme im Zusammenhang mit linearen Berechnungen auf Grund der vom SYSIN-Modul gespeicherten Daten. Dabei geht es um die Lösung linearer inhomogener Gleichungssysteme für statische Probleme und um die Bestimmung von Eigenwerten und Eigenvektoren für dynamische modale Analysen oder für Stabilitätsprobleme. Inhomogene Gleichungssysteme werden mit Hilfe einer aus [5] übernommenen «Skyline»-Routine mit sehr grosser Kapazität gelöst («Substructuring» ist jedoch im FLOWERS nicht möglich). Die gleiche Routine wird auch zur Lösung von Eigenwertproblemen nach der Methode der Unterraumiteration (siehe z.B. [6]) verwendet. Zur Verkleinerung des Rechenaufwandes kann die interne Knotennumerierung vom Programm optimiert werden [7].

STOUT dient bei statischen Problemen zur numerischen Ausgabe von Knotenverschiebungen, Auflagerkräften und Elementspannungen. Diese werden von entsprechenden Elementroutinen auf Grund der vom SOLVE-Modul erhaltenen Lösungsvektoren ermittelt.

DNOUT dient bei dynamischen Problemen zur Berechnung und zur numerischen Ausgabe sowohl des zeitlichen Verlaufs für eine spezifizierte Lastgeschichte als auch der Maximalwerte nach der Methode der Antwortspektren von Verschiebungs-, Geschwindigkeits-, Beschleunigungs-, Auflagerkraft- und Spannungskomponenten.

NONBA und NONIN dienen zur Durchführung sowohl statischer nichtlinearer Analysen nach dem modifizierten Newton-Raphson-Verfahren als auch dynamischer nichtlinearer Analysen nach verschiedenen impliziten und expliziten Algorithmen. Bemerkenswert ist dabei die Möglichkeit, den für solche Probleme notwendigen numerischen Iterationsprozess während der Programmausführung vom Bildschirmterminal aus zu verfolgen und zu steuern. Darüber informiert im Detail der vor kurzem erschienene Artikel [4].

GRAPH dient zur graphischen Ausgabe verschiedener Problem- und Resultate. Mehr darüber im folgenden Abschnitt.

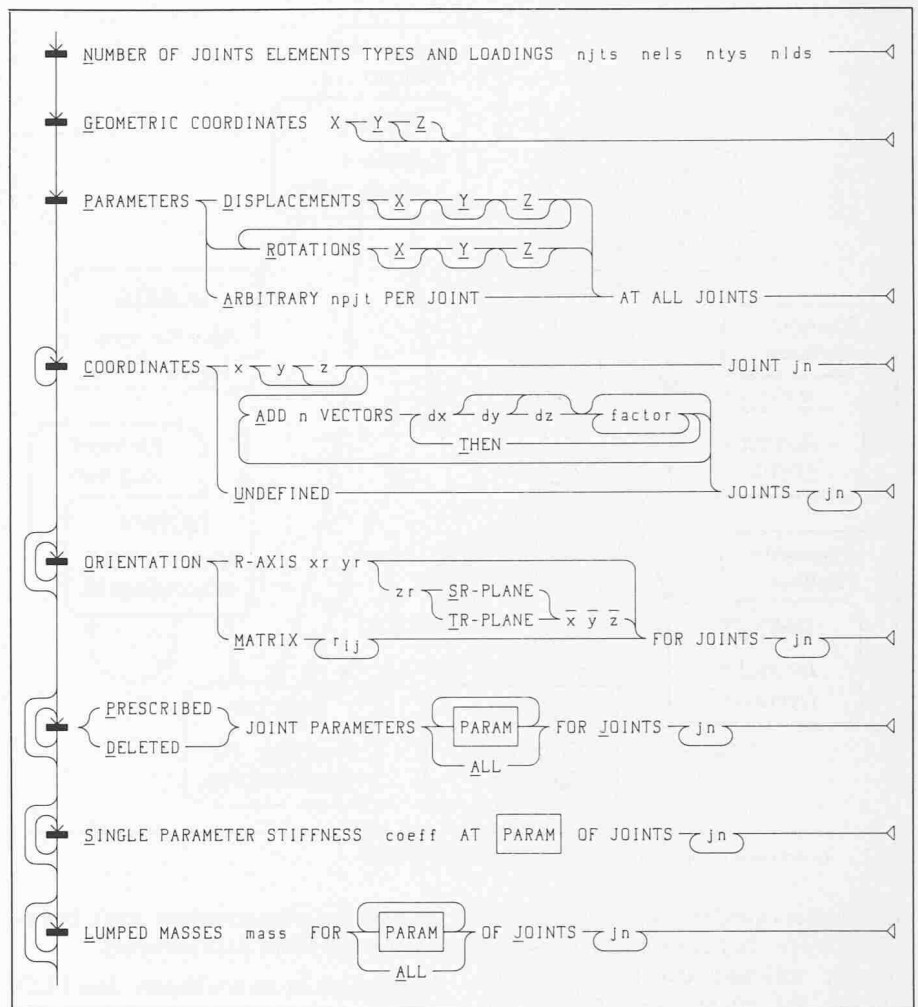


Bild 1. Module des Programms FLOWERS

POSTP dient zur Erzeugung einer besonders formatierten Datei, die von externen Nachlaufprogrammen möglicherweise auf fremde Hardware leicht gelesen und weiterverarbeitet werden kann. Diese Datei kann jeweils verschiedene, vom Programm benutzer ausgewählte Problem- und Resultate enthalten (das Modul POSTP ist z.Z. allerdings noch in der Planungsphase).

Ein- und Ausgabe

Alle Module des Programms FLOWERS, mit Ausnahme des für nichtlineare Analysen verwendeten Moduls NONIN, lesen ihre Eingabedaten von einer im voraus vorbereiteten Eingabedatei. Sie arbeiten also im «Batch-Betrieb». Nach unserer Ansicht ist dies sinnvoll, da eine interaktive Spezifizierung der Problemausgangsdaten und der gewünschten Resultate im Dialog mit dem Computer für lineare FE-Anwendungen mehr Nachteile als Vorteile hätte. Selbstverständlich ist aber beim FLOWERS, so wie bei anderen FE-

Grossprogrammen, die Generierung der Elementmasche mit Hilfe interaktiver graphischer Vorlaufprogramme, möglicherweise im Zusammenhang mit übergeordneten CAD-Systemen, sinnvoll und empfehlenswert. Die Vorbereitung der Eingabedaten erfolgt, ähnlich wie beim Statik-Programm, auf Grund einer formatfreien, problemorientierten Eingabesprache, die in der Benutzeranleitung des FLOWERS-Programms mit Hilfe weniger Syntaxdiagramme beschrieben ist (siehe «FLOWERS User's Manual» [8], eine zweite Auflage davon wird im Frühling 1983 veröffentlicht). Bild 2 zeigt einen Ausschnitt des Syntaxdiagramms zur Beschreibung der Eingabe des Moduls SYSIN. Die auf einigen einfachen Konventionen basierenden Syntaxdiagramme definieren im Detail die Syntax der Eingabesprache und zeigen in kompakter und übersichtlicher Art und Weise die vom Programm erwarteten Daten und damit auch viele der im Programm eingebauten Möglichkeiten. Wie schon beim STATIK-Programm haben sie sich auch für die Benützung des FLOWERS-Programms als sehr benutzerfreundliches Hilfsmittel erwiesen.

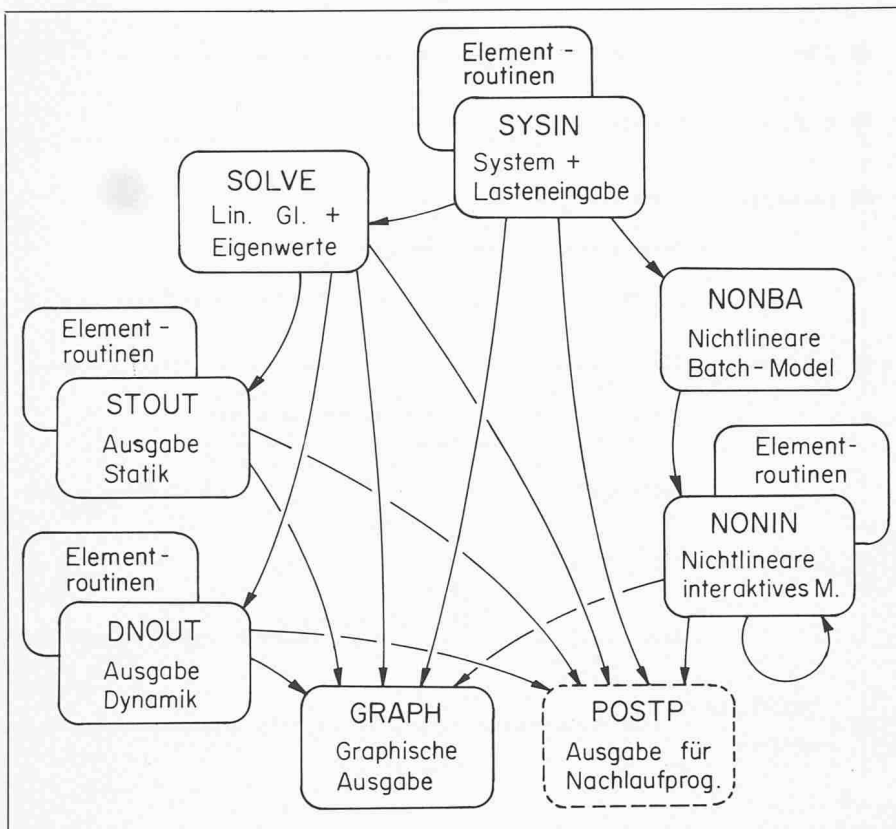


Bild 2. Ausschnitt aus dem Syntaxdiagramm des Moduls SYSIN

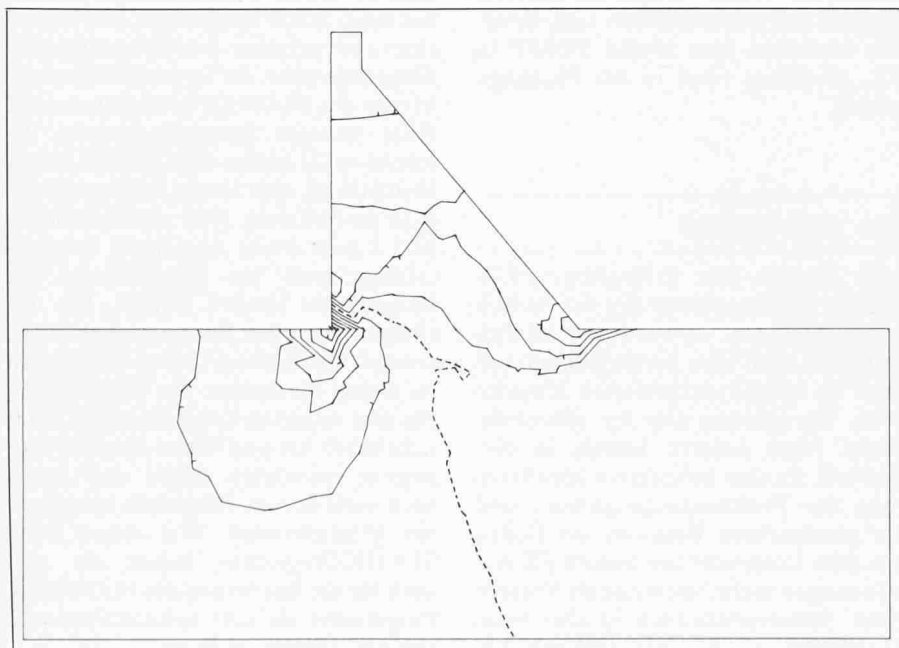
Ganz anders arbeitet hingegen das Modul NONIN. Dabei erfolgen Ein- und Ausgabe während der Programmausführung auf einem gewöhnlichen alphanumerischen Bildschirmterminal in «Full-Screen»-Modus, womit eine direkte Interaktion zwischen Programm- benützer und Computer möglich ist. Unserer Ansicht nach ist dies für nicht-lineare Probleme sinnvoll, da der numerische Iterationsprozess von verschiedenen, im voraus schwer festzule-

genden Steuerparametern stark beeinflusst wird. Mehr darüber in [4].

Schliesslich ist zu erwähnen, dass FLOWERS verschiedene Daten und Resultate auch in graphischer Form ausgeben kann. Dafür ist das Modul GRAPH zuständig, das folgende Arten von Bildern erzeugen kann:

- Elementmasche oder Teile davon in axonometrischer Darstellung.
- Verformte Elementmasche oder Teile

Bild 3. Isoliniendarstellung der vertikalen Normalspannungen infolge Wasserdruck bei einer Gewichtstau-mauer



davon ebenfalls in axonometrischer Darstellung. Die dabei verwendeten Knotenverschiebungen können sowohl aus der Lösung statischer oder dynamischer linearer oder nichtlinearer Probleme als auch aus Eigenvektorenberechnungen im Zusammenhang mit dynamischen oder Stabilitätsanalysen stammen. Es ist zudem möglich, irgendein Knotenparameter (z.B. die Temperatur bei Wärmeleitungsproblemen) als Verschiebung in einer bestimmten Koordinatenrichtung darzustellen. Durch die Erzeugung mehrerer solcher Bilder für verschiedene Laststufen oder Zeitschritte kann eine Art Trickfilm hergestellt werden.

- Isoliniendarstellung bestimmter elementinterner, in den Knoten gemittelter Spannungsponenten. Ein Beispiel davon wird im Bild 3 gezeigt.
- Diagrammzeichnungen bestimmter Variablen (Verschiebungen, Auflagerkräfte, Spannungsponenten usw.) als Funktion sei es der Zeit, sei es irgendeiner anderen, vom Programm- benützer gewählten Variable.

Das Modul GRAPH erzeugt jeweils eine besondere Datei, die unabhängig vom verwendeten graphischen Ausgabegerät formatiert ist. Zur Erzeugung von Bildern müssen folglich die in dieser Datei enthaltenen Bilddaten von einem Nachlaufprogramm gelesen und in geräteabhängige Steuerbefehle umgesetzt werden, womit Kompatibilität mit jedem Ausgabegerät gewährt wird. (Heute existieren Nachlaufprogramme für die Tektronix 4014 und für Calcomp- und Benson-Plotter. Nachlaufprogramme für andere Geräte wären jedoch leicht zu schreiben.) Dies ermöglicht auch noch, für die FLOWERS-Berechnungen und für die Bild- erzeugung zwei verschiedene Computer, z.B. einen Rechenzentrum-Computer und einen eigenen Mikrocomputer, zu verwenden.

Elementbibliothek

Die möglichen Anwendungsbereiche des Programms FLOWERS hängen von den zur Verfügung stehenden Element- routinen ab, welche die Elementbibliothek bilden. Sie umfasst heute 19 lineare und 8 nichtlineare Elementmodelle (oder kurz «Elemente»), welche die Lösung einer Reihe verschiedener Probleme erlauben. Zu jedem Element gehören einige Elementroutinen, die von den Systemroutinen bestimmter Programm- module aktiviert (oder «angerufen») werden und die folgende Aufgaben zu übernehmen haben:

1. Interpretation, Kontrolle und Echo-print der vom Programm benutzer spezifizierten, elementeigenen Eingabedaten. Die syntaktische richtige Reihenfolge und die physikalische Bedeutung der von jedem Bibliothekselement erwarteten Eingabedaten werden, wie bei den Systemdaten, mit Hilfe übersichtlicher Syntaxdiagramme beschrieben.
2. Bildung der für lineare Analysen notwendigen Elementmatrizen, aus denen die Beiträge jedes Elementes zum globalen System numerisch erfasst werden (Steifigkeits-, geometrische Steifigkeits-, Massen-, Last- und Spannungsmatrizen).
3. Berechnung bestimmter, je nach Element verschiedener Spannungskomponenten in einigen Punkten des Elementes (meistens in der Elementmitte und in den Elementecken oder in den Integrationspunkten). Dazu werden sowohl die Lösungsvektoren als auch die früher von anderen Routinen des gleichen Elementes bestimmten und vom System gespeicherten Spannungsmatrizen verwendet.
4. Nur für nichtlineare Elemente: Bestimmung der Knotenkräfte und der aktuellen tangentialen Steifigkeitsmatrix für gegebene Knotenverschiebungen. Dabei ist zu bemerken, dass bei nichtlinearen Problemen die Knotenkräfte und die tangente Steifigkeitsmatrix meistens Funktionen der Verformungsgeschichte sind. Dies bedingt, dass im Laufe der Analyse elementeigene, «historische» Daten, welche von den Elementroutinen gebildet und geändert werden, vom System gespeichert und verwaltet werden müssen.

Aus diesen Ausführungen ist ersichtlich, dass die System- und die Elementroutinen, die in der Regel von verschiedenen Programmierern geschrieben werden, sehr eng zusammenarbeiten müssen, wofür saubere und leichtverständliche Schnittstellenkonventionen notwendig sind. Weil FLOWERS ein *Forschungsinstrument*, und zwar vor allem *auf der Stufe der Elementmodelle*, werden soll, wurde dieser Frage grosse Aufmerksamkeit geschenkt.

Die Elementbibliothek ist damit ohne prinzipielle Schwierigkeiten beliebig

erweiterbar. Für lineare statische und dynamische Analysen stehen heute ebene und räumliche Fachwerk- und Rahmenelemente sowie Scheiben-, Membran-, Platten-, ebene Schalen-, rotationssymmetrische und dreidimensionale isoparametrische Elemente mit variabler Anzahl Knoten (d.h. mit oder ohne Seitenknoten) und verschiedenen Elementlasten zur Verfügung. Zudem hat man zweidimensionale isoparametrische, dreieckige und viereckige Elemente zur Lösung von Feldproblemen, wie sie sich z.B. bei Wärmeleitungsrechnungen oder bei elektrischen oder magnetischen Fehlberechnungen stellen.

Für nichtlineare statische Berechnungen nach der modifizierten Newton-Raphson-Methode und für nichtlineare dynamische Berechnungen nach verschiedenen impliziten und expliziten Algorithmen unter Berücksichtigung sowohl geometrischer als auch materialbedingter Nichtlinearitäten stehen heute Fachwerk-, Rahmen-, Kontakt-, Platten- und ebene Schalenelemente zur Verfügung. Andere lineare und nichtlineare Elementmodelle sind in Vorbereitung.

Schlussbemerkungen

Gestützt auf langjährige Erfahrungen bezüglich der Entwicklung von FE-Programmen hat man mit dem System FLOWERS versucht, ein Instrument der Hochschullehre, der Hochschulforschung sowie auch der Praxis für die Bau- und die Maschinenindustrie zu schaffen. Teile des Programms werden seit zwei Jahren zu Übungszwecken von den Bauingenieurstudenten der ETH Zürich verwendet. Heute stellt sich allerdings die *Frage nach der Verfügbarkeit des Programms für ein breites Spektrum potentieller Benutzer* sowie die damit zusammenhängende und zur Zeit eigentlich im Vordergrund stehende Frage der *Benützerausbildung*. Dafür wird im Laufe des Sommersemesters 1983 der Fortbildungskurs «Lineare und nichtlineare Finite-Element-Methoden» veranstaltet, bei dem Assistenten und wissenschaftlichen Mitarbeitern der ETH Zürich sowie Bau- und Maschineningenieuren aus der Praxis die Möglichkeit geboten wird, neben

Literatur

- [1] Anderheggen, E., Alberti, G., Lässker, A. (1972) «Handbuch der STRESS-Sprache», Institut für Baustatik und Massivbau, ETHZ, 3. Auflage
- [2] Anderheggen E., Green, D., Walder, U. (1976): «Das Computerprogramm FLASH», Institut für Baustatik und Konstruktion, ETHZ, Bericht Nr. 64
- [3] Anderheggen, E., Bazzi, G., Müller, P., Theiler, J., Walder, U. (1983) «STATIK-Programm: Benutzeranleitung», Institut für Informatik, ETHZ, 3. Auflage
- [4] Anderheggen, E. (1983): «Nichtlineare Finite-Element-Berechnungen und Informatik», Symposium über angewandte Mechanik in Industrie und Hochschule, Schweizer Ingenieur und Architekt, Heft 4, 42-46
- [5] Monk, D.P., Powell, G.H. (1974): «Large Capacity Equation Solver for Structural Analysis», Int. J. for Computer and Structures, Vol. 4, 699-728
- [6] Bathe, K.J., Wilson, E.L. (1976): «Numerical Methods in Finite Element Analysis», Prentice-Hall
- [7] Gibbs, N.E. et al. (1976): «Matrix Bandwidth and Profile Reduction», ACM Transactions on Mathematical Software, Vol. 2, No. 4
- [8] Anderheggen, E., Bazzi, G., Elmer, H., Friedrich, I., Maag, H., Theiler, J. (1981): «FLOWERS User's Manual», Institut für Informatik, ETHZ

der Behandlung theoretischer Fragen auch noch direkte Erfahrungen bei der praktischen Anwendung von FE-Methoden zu sammeln. Dies soll mit Hilfe des gerade für solche Zwecke geplanten Programms FLOWERS erfolgen, wobei jedoch zu bemerken ist, dass der im Fortbildungskurs behandelte Stoff eine notwendige Voraussetzung zur Verwendung jedes FE-Programms darstellt. Später soll das Programm FLOWERS bei verschiedenen Lehrveranstaltungen, u.a. an der neuen *Abteilung für Informatik* der ETH Zürich verwendet werden.

Die Entwicklung des Programms FLOWERS stellt eine *Gruppenarbeit* dar. Die von den Mitarbeitern des Verfassers, Dr. G. Bazzi, H. Elmer, T. Friedrich, H. Maag, J. Theiler und Dr. U. Walder geleisteten Beiträge verdienen besondere Anerkennung.

Adresse des Verfassers: Prof. Dr. E. Anderheggen, Institut für Informatik, ETH-Hönggerberg, 8093 Zürich.