

Zeitschrift: Schweizerische Bauzeitung
Herausgeber: Verlags-AG der akademischen technischen Vereine
Band: 85 (1967)
Heft: 21

Artikel: Die problemorientierte Computersprache STRESS in der Bauingenieurpraxis
Autor: Mauch, S.P.
DOI: <https://doi.org/10.5169/seals-69461>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. 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. [Siehe Rechtliche Hinweise.](#)

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. [Voir Informations légales.](#)

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. [See Legal notice.](#)

Download PDF: 14.03.2025

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

Einleitung

Der Einsatz von elektronischen Datenverarbeitungsanlagen für Probleme des Bauingenieurs ist in den letzten drei Jahren durch die Einführung von problemorientierten Computersprachen wesentlich beeinflusst worden. Dieser Einfluss hat seinen Niederschlag gefunden in der kürzlich abgehaltenen vierten Konferenz über Elektronische Datenverarbeitung der American Society of Civil Engineers [1]. Ziel des vorliegenden Beitrages ist es, am Beispiel von STRESS¹⁾ — eines der auch bei uns bekanntesten POL-Systeme²⁾ — die Bedeutung und die Anwendungsbereiche dieser problemorientierten Computersprachen für die Praxis darzustellen. Das angeführte Beispiel eines mit dem Bodenmaterial zusammenwirkenden Tunnelgewölbes soll die Vielfältigkeit der Anwendungsmöglichkeiten illustrieren.

1. Entwicklung und Hierarchie der Computersprachen

Computersprachen bilden das geistige Kommunikationsmedium zwischen Mensch und Computer. Während diese Programmiersprachen am Anfang noch ganz maschinenorientiert waren, haben sie sich im Laufe der Zeit durch die Entwicklung in drei bedeutenden Schritten immer mehr den Bedürfnissen und der Sprache des Menschen angepasst. Es hat sich damit eine Hierarchie der Computersprachen gebildet, wie sie in Tabelle 1 dargestellt ist. Für jede Sprachstufe ist ein Beispiel eines Befehls gezeigt. Während dieser Befehl in den Stufen 1 und 2 lediglich eine einzige Addition bewirkt, beinhaltet der Befehl der Stufe 3 (Formelsprachen) eine Reihe von algebraischen Operationen. Der einzige Befehl PRINT FORCES 1 TO 30 kann das Ausdrucken der Stabkräfte an den Stäben 1 bis 30 eines statischen Systems auslösen.

Der Beginn der eigentlichen Anwendungen von Computern auf Bauingenieurprobleme fällt in die Zeit der ersten Versionen der Formelsprachen (Stufe 3 in Tabelle 1) FORTRAN und später ALGOL. Diese haben es nach einer zehnjährigen Periode des maschinenorientierten Programmierens (Stufen 1 und 2 in Tabelle 1) ermöglicht, plötzlich mit einem viel kleineren Programmieraufwand dem Computer mathematische Prozeduren verständlich zu machen. Es ist deshalb natürlich, dass in der Folge die Bauingenieure mit diesem neuen Hilfsmittel ihre mathematischen Berechnungen, wie die Lösung von algebraischen Gleichungssystemen in der Baustatik oder von Differentialgleichungen in der Hydraulik, zu programmieren begannen. Die eigentliche Bedeutung dieser Entwicklung war, dass es möglich wurde, komplizierte mathematische Probleme zu lösen, die bis anhin wegen des grossen Rechenaufwandes gar nicht zu bewältigen gewesen waren.

In der Folge hat eine rund achtjährige Erfahrung in der Programmierung von Ingenieurproblemen mit diesen Formelsprachen (oder Compilersprachen) auf eine Reihe von neuen Begrenzungen geführt, welche einer Anwendung des Computers in sehr vielen Nicht-Spezialfällen aus Wirtschaftlichkeitsgründen im Wege standen. Diese Begrenzungen könnten wohl unter dem Begriff *Sprach- oder Kommunikationsbarriere* zusammengefasst werden. Die vorhandenen Computersprachen waren den spezifischen Bedürfnissen der Ingenieure noch ungenügend angepasst, um eine rasche Programmierung zu gestatten. Die Beseitigung dieser Sprachbarriere hat zum Konzept der problemorientierten Computersprachen — die der vierten der in Tabelle 1 dargestellten Stufen angehören — geführt.

¹⁾ STRESS = Structural Engineering Systems Solver

²⁾ POL = Problem Oriented Language

Tabelle 1. Die Hierarchie der Computersprachen

Stufe	Sprache	Befehlsform
1	Absolute Maschinsprache	00057 0500 0 77463
2	Symbolische Maschinsprache (z. B. FAP, CODAP, SPS)	START CLA RHO
3	Formelsprachen (z. B. FORTRAN, ALGOL)	A = REX * (B — 1)
4	Problemorientierte Sprachen (z. B. STRESS, COGO, STRUDL)	PRINT FORCES 1 TO 30

2. Konzept der problemorientierten Computersprache

Der Übergang von der niedrigeren zur höheren Stufe der Programmiersprache bedeutet einerseits eine fortschreitende Automatisierung und Leistungssteigerung in der Programmierung und andererseits eine zunehmende Spezialisierung auf ein bestimmtes Anwendungsgebiet. Daraus ergibt sich die Grundidee der problemorientierten Computersprache: Die Beschränkung in der Programmierung auf ein bestimmtes Fachgebiet ermöglicht eine bedeutende Leistungssteigerung, indem die Kommunikationsbarriere Mensch-Computer beseitigt wird.

Ein problemorientiertes Programmiersystem besteht aus einer Sprache und einem Prozessorprogramm. Diese Sprache enthält ein Vokabular von Fachausdrücken aus dem betreffenden Fachgebiet und wird durch die interpretierende Funktion des Prozessorprogrammes dem Computer verständlich gemacht. Verschiedene Ausdrücke können zu problemorientierten Befehlen zusammengefasst werden. Eine sinnreiche Folge von solchen Befehlen ergibt ein problemorientiertes Quellenprogramm. Weil dieses Programm aus Ausdrücken der entsprechenden Fachsprache besteht, kann es auch ohne weiteres in sehr kurzer Zeit aufgestellt und sofort von anderen Sachverständigen des Gebietes verstanden werden. Ebenfalls kann natürlich eine solche Computersprache von Fachleuten in kürzester Zeit erlernt werden.

Ein Quellenprogramm wird im Computer vom Prozessorprogramm interpretiert und die entsprechenden Aufgaben direkt ausgeführt. Der Prozessor übersetzt also das Quellenprogramm nicht jedesmal in ein neues Maschinenprogramm (wie dies bei einem Compiler geschieht), sondern er setzt aus der Interpretation des Quellenprogrammes eine Folge von vorprogrammierten Maschinenprogrammblöcken zusammen und führt diese direkt aus. Deshalb kann man einen solchen Prozessor einen Pseudo-Compiler nennen. Ein einziger problemorientierter Befehl, wie z. B. der SOLVE THIS PART-Befehl von STRESS kann Tausenden von Maschineninstruktionen entsprechen und besitzt deshalb einen viel grösseren Informationsinhalt oder Aussagekraft als ein FORTRAN-Befehl. Daraus resultiert die erwähnte Automatisierung in der Programmierung von Ingenieurproblemen.

3. STRESS als Beispiel einer problemorientierten Computersprache für Bauingenieure

Die Forschung über Probleme der Kommunikation zwischen Bauingenieur und Computer hat ungefähr im Jahre 1960 am Massachusetts Institute of Technology (MIT) in den USA begonnen und hat in einer frühen Phase zur Entwicklung der ersten solchen problemorientierten Sprache, COGO, geführt [2]. COGO wurde geschrieben für das Gebiet der ebenen und räumlichen Geometrieaufgaben des Vermessungswesens und des Strassenbaues. In einer zweiten und umfassenderen Phase ist in den Jahren 1963 bis 1964 auch am MIT das STRESS-System für das Gebiet der Baustatik entwickelt worden [3] [4].

Seither sind aufgrund der Forschungen, welche zu COGO und STRESS geführt haben, die Anstrengungen des MIT auf die Entwicklung eines integrierten Programmiersystems unter dem Namen ICES (Integrated Civil Engineering System) konzentriert worden [5]. Innerhalb von ICES wird es möglich sein, eine beliebige Anzahl von Untersystemen wie COGO, STRESS usw. in einheitlicher Form unter dem ICES-Monitor zu verwenden. Zudem wird ICES die Entwicklung von solchen Untersystemen bedeutend erleichtern, indem es für die Programmierung der Prozessorprogramme der Untersysteme selber problemorientierte Befehle enthält.

Sowohl STRESS als auch COGO sind in den letzten Jahren in der Praxis auf breiter Basis verwendet worden. Sie sind bis heute auf einer Anzahl von Computern verschiedener Herstellerfirmen übertragen worden und in Gebrauch. Es hat sich gezeigt, dass das Konzept der Computeranwendung mit problemorientierten Sprachen einen zunehmenden Anwendungsbereich findet. Eine nähere Beschreibung der STRESS-Sprache an dieser Stelle erübrigt sich, da diese im Detail in [3] und in der deutschen Literatur generell in [6, 7] beschrieben ist. Es sollen hier die mit STRESS gemachten Erfahrungen dargestellt und die POL-Methode mit andern Methoden der Computeranwendung für Ingenieurprobleme verglichen werden.

Die Zielsetzung bei der Entwicklung des STRESS-Systemes war die grundsätzliche Verbesserung der *Kommunikation zwischen Bau-*

ingenieur und Computer und besteht im wesentlichen aus folgenden drei Punkten:

1. Beseitigung der Sprachbarriere durch Schaffung einer dem Benutzer angepassten Computersprache.
2. Schaffung eines Systemes, welches mit dem iterativen Charakter der Ingenieurprobleme verträglich ist. Die Resultate einer statischen Berechnung eines Bauwerkes sollen für eine anschließende Bemessung erhalten bleiben und umgekehrt, so dass zum Beispiel eine erneute Berechnung des selben Systemes möglich ist, ohne dass alle Rohdaten neu gegeben werden müssen.
3. Schaffung eines offenen, dynamischen Systemes, welches nach Belieben in seinen Fähigkeiten erweitert werden kann. Zu diesem Zweck ist eine strenge logische Gliederung (Modularität) der Prozessor-Programmteile nach dem Baukastenprinzip notwendig. Nur durch eine solche Gliederung wird es je möglich sein, die Voraussetzungen für ein zusammenwirkendes System zwischen Benutzer und Computer auf einer Timesharingbasis zu schaffen. Inwiefern dieses letzte Ziel schon in STRESS erreicht worden ist, ist von S.J. Fenves [8] beschrieben worden.

Das STRESS-System hat sich zuerst einmal hauptsächlich mit der Analyse in der Baustatik und nur in beschränktem Masse mit der Bemessung der Tragwerke befasst. Dadurch konnten die Hauptanstrengungen dazu benutzt werden, ein für den Ingenieur praktisch brauchbares, flexibles und ausbaubares Grundsystem zu entwickeln. Das aus STRESS weiterentwickelte System STRUDL umfasst sowohl analytische als auch Bemessungs-Prozeduren [9].

Zu der Vorstellung eines zusammenwirkenden Systemes Mensch-Computer sind Datenfernübertragung und Timesharing wesentliche Bestandteile. Deshalb ist das ganze STRESS-System grundsätzlich timesharing orientiert, und die interpretierenden und iterativen Befehle in STRESS, wie

SELECTIVE OUTPUT	(Selektive Ausgabe)
PRINT	(Ausgabe auf Drucker)
MODIFICATIONS	(Varianten)
ADDITIONS	(Zusätze)
DELETIONS	(Weglassungen)
CHANGES	(Abänderungen)

gewinnen erst in diesem Zusammenhang ihre volle Kraft und Bedeutung [7]. Jedoch sind sie auch im gewöhnlichen Batchmodus, wo keine direkte Konversation mit dem Computer vorhanden ist, äusserst nützlich, sobald ein statisches System nicht nur in einer festen Form analysiert, sondern vom Ingenieur nach verschiedenen Gesichtspunkten untersucht werden muss. Dieser Fall ist in der Praxis der häufigste und wird anschliessend an einem Beispiel erläutert.

Obwohl in Zukunft der Computereinsatz immer mehr auf der Timesharing-Basis erfolgen wird, hat die konventionelle Anwendung im Batchmodus trotzdem weiterhin ihre Berechtigung für gewisse Aufgaben. In analoger Weise wird es neben dem Gebrauch von *problemorientierten Programmiersystemen* auch weiterhin Anwendungen von *konventionellen Bibliothekprogrammen* geben.

Letztere Methode wird sich in Zukunft dort weiter lohnen, wo bestimmte, sich in relativ engem Rahmen wiederholende Aufgaben immer wieder gleich, aber mit neuen numerischen Daten, zu lösen sind. Generell gesehen werden diese Anwendungen hauptsächlich dort stattfinden,

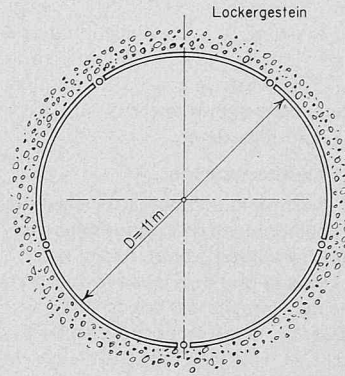
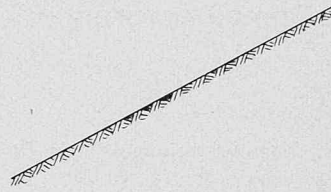


Bild 1. Querschnitt des Tunnelgewölbes

den, wo die Informationsübertragung Mensch-Computer eine untergeordnete Rolle spielt und die Eingabe wenige, hauptsächlich numerische Daten, umfasst. Ein typisches Beispiel findet man in den Lohn- oder Buchhaltungsabrechnungen. Ganze Ingenieurprobleme sind aber immer Einzelfälle, also nie repetitiv. Es lässt sich jedoch bei der Zerstückelung in untergeordnete Teilprobleme oder Prozeduren eine Schwelle feststellen, unterhalb welcher auch Teile von Ingenieurproblemen als repetitiv bezeichnet werden können. Die problem- und kommunikationsorientierte Computer-Anwendung eignet sich vor allem für Probleme oberhalb dieser Schwelle, während unterhalb dieser Schwelle oft der Einsatz von vorhandenen Standardprogrammen wirtschaftlicher ist.

4. Praktisches Anwendungsbeispiel mit STRESS

Das im folgenden beschriebene Problem, welches mit STRESS behandelt wurde, ist nicht in allen Aspekten ein typisches Beispiel für die routinemässige Anwendung einer problemorientierten Sprache, indem es — weniger im Vorgehen bei der Analyse und Bemessung — sondern in seinem physikalischen Aufbau, nicht ein Routineproblem ist. Es kommt aber trotzdem zum Ausdruck, wie sehr die problemorientierte Sprache und die Modifikationsmöglichkeiten die Arbeit des Ingenieurbüros mit dem Computer erleichtern können. Es handelt sich um ein kreisförmiges Stahlbetongewölbe im Lockergestein. Das ringförmige Tunnelgewölbe besteht aus fünf Teilen, welche durch gelenkige Fugen verbunden sind. Die Fugen können keine Biegemomente, wohl aber Quer- und Normalkräfte übertragen. Der Querschnitt des Tunnels ist in Bild 1 dargestellt.

Die Schwierigkeit bei der Untersuchung des als zweidimensional betrachteten Systemes besteht offenbar im Zusammenwirken zwischen Boden und Einbau. Die resultierenden Erddrücke auf dem Gewölbe

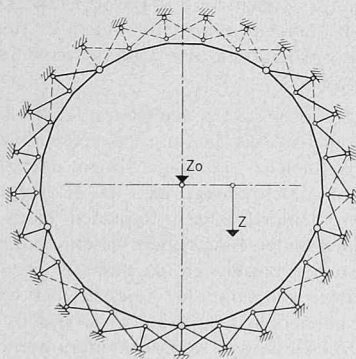
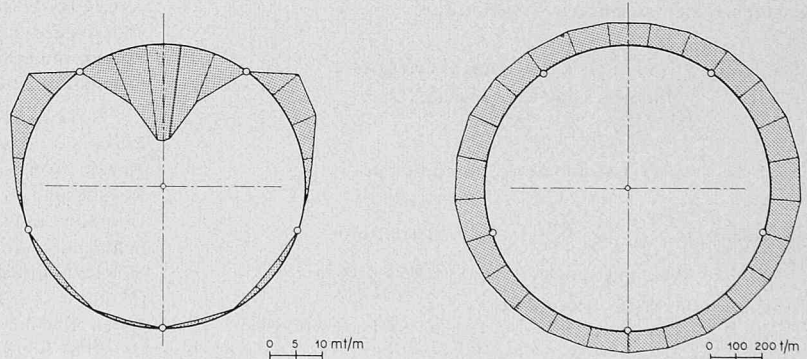


Bild 2. Gedankenmodell des Systemes Einbau-Boden (die Zugstäbe sind gestrichelt eingezeichnet)



a) Biegemomente
b) Normalkräfte
Bild 3. Schnittkräfte bei horizontaler Geländelinie und sattem Einbau

sind nicht bekannt, sondern verformungsabhängig, was beim vorhandenen Beispiel des 5-Gelenk-Bogens sehr deutlich zum Ausdruck kommt: Ohne die stützende Wirkung des seitlich umgebenden Erdmaterials wäre das Tragwerk instabil. Wenn sowohl der Einbau als auch das Bodenmaterial als linear elastische Medien modelliert werden, so wird die Gleichgewichtslage und damit die Kräfteverteilung durch die Elastizitätskonstanten des Bodens und des Einbaues beeinflusst.

Nach ähnlichen Überlegungen wie sie bei den Methoden der «finite elements» üblich sind, wurde die den Einbau umgebende elastische Ebene durch äquivalente Stäbe und der Einbau selber durch ein vielseitiges Polygon modelliert. In erster Näherung wurde das in Bild 2 dargestellte Modell gewählt. Die Eigenschaften der radialen und diagonalen Bodenstäbe können aus den elastischen Konstanten des Bodens (E -Modul und Querdehnungszahl, respektive «Bodensteifigkeit») abgeleitet werden. Damit ergibt sich ein ebenes elastisches Stabwerk, bestehend aus gelenkigen und biegesteifen Stäben (Bild 2). Die vernachlässigbare Zugfestigkeit des Bodens wird durch Elimination jener Bodenstäbe, welche eine Zugkraft aufweisen, berücksichtigt. Dieses System kann nun für irgendeinen Belastungsfall mit Hilfe der STRESS-Sprache mit Leichtigkeit beschrieben werden. Als erster Schritt werden jedoch die am Modell anzubringenden Belastungen bestimmt. Unter der idealen, aber für den Einbau ungünstigen Annahme, dass der Tunneleinbau satt in das ungestörte Gebirge eingebaut werden kann, sind diese Belastungen gerade die Ruhespannungen des elastischen Gebirges. Sie greifen am Umfang des Gewölberinges an und betragen für den Fall einer horizontalen Oberfläche:

$$\begin{aligned} \text{vertikal} \quad \sigma_v &= \gamma (z_0 + z) \\ \text{horizontal} \quad \sigma_h &= \lambda_0 \gamma (z_0 + z) \end{aligned}$$

wo λ_0 die Ruhedruckziffer darstellt. Die Belastungen werden (hier aus Gründen der numerischen Genauigkeit) nach der üblichen Methode in den isotropen und anisotropen Belastungsteil aufgespalten. Sie werden an den mit den Bodenstäben gelenkig verbundenen Ringstäben angebracht. Damit ergeben sich an der Fläche zwischen Einbau und Boden folgende Belastungsspannungen σ_v in vertikaler und σ_h in horizontaler Richtung:

$$\begin{aligned} \text{isotrop} \quad \sigma_v = \sigma_h &= \lambda_0 \gamma z_0 \\ \text{anisotrop} \quad \sigma_v(z) &= \gamma z + (1 - \lambda_0) \gamma z_0 \\ \sigma_h(z) &= \lambda_0 \gamma z \end{aligned}$$

Dabei bedeutet z_0 die Überlagerungshöhe des Kreismittelpunktes und ($z_0 + z$) diejenige irgendeines Punktes. Nur der anisotrope Teil erzeugt Biegemomente im Gewölbe, während aber beide Belastungsanteile Normalkräfte erzeugen. Die Schnittkräfte im Gewölbe werden um so kleiner, je steifer der Boden im Verhältnis zum Einbau ist. Die fünf Gelenke und die grosse Schlankheit des Einbaues machen diesen extrem biegeweich, so dass nur relativ kleine Biegemomente auftreten und die Normalkraft die überwiegende Beanspruchung darstellt.

Die Schnittkräfteverteilungen für den Fall ohne anfänglichen Zwischenraum hinter dem Tübbing sind in Bild 3 schematisch dargestellt. Diese sind mit STRESS, ausgehend von dem in Bild 2 dargestellten statischen System, in zwei Schritten ermittelt worden. Nach der ersten Analyse wurden diejenigen Bodenstäbe, welche total eine Zugkraft aufwiesen, mit Hilfe der STRESS-Modifikationsbefehle eliminiert und das neue System wieder gelöst. Auf diesem Weg wurde die verschwin-

dende Zugfähigkeit des rolligen Bodens in Betracht gezogen, und diese Nichtlinearität der Eigenschaften des Bodenmaterials simuliert.

Das Problem konnte in der Folge mit Hilfe von weiteren Modifikationen nach der Auswirkung von anderen Parametervariationen wie Änderung der Bodensteifigkeit oder Einführung eines anfänglichen Hohlraumes hinter dem Einbau untersucht werden. Diese Modifikationen umfassten Änderungen in den Stabeigenschaften, in den Belastungen, in der Geometrie, sowie in den Randbedingungen des Systemes und konnten mit Leichtigkeit vorgenommen werden. Zum Beispiel können für die Elimination der Zugstäbe 30, 16, 20, 14 aus dem Stabsystem die folgenden STRESS-Befehle genügen:

DELETIONS
MEMBERS 30, 16, 20, 14

In Bild 4 sind noch die Schnittkräfte im Gewölbe bei geneigter Geländeoberfläche dargestellt. In diesem Fall ergibt sich durch die unsymmetrische Belastung eine entsprechende Kräfteverteilung. Wieder ist ersichtlich, dass sich die Biegemomente am Einbau sehr bescheiden ausnehmen. Der Grund dazu liegt in der grossen Flexibilität des Ringes, welcher relativ grosse Bewegungen ohne Zerstörung mitmachen kann und so das umgebende Gebirge vermehrt zur Lastübernahme zwingt.

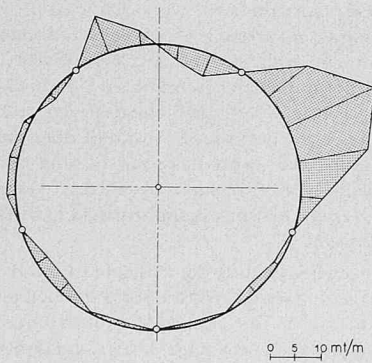
Wegen der grossen Flexibilität des Systems gilt das Interesse nicht nur den Schnittkräften im Gewölbeinbau, sondern ebenso sehr den zu erwartenden Deformationen. Während die Berechnung der Deformationen normalerweise einen weiteren und zudem recht mühsamen Rechnungsgang darstellt, so können mit der STRESS-Sprache die Verschiebungen einzelner oder aller Knotenpunkte gleichzeitig mit den Schnittkräften erhalten werden. Für den Fall der geneigten Geländeoberfläche sind die elastischen Deformationen in Bild 5 dargestellt.

Ein weiterer Vorteil der Anwendung des STRESS-Systems war bei den Untersuchungen des Gewölbes die Leichtigkeit, mit der irgendeine der verschiedenen Fragestellungen dem Computer verständlich gemacht werden konnte, was sich durch die Zweckmässigkeit der Sprache und die Allgemeinheit des STRESS-Systems ergibt. Der Zweck der Untersuchungen war nicht die komplette Ermittlung der auftretenden Beanspruchungen, sondern das Erarbeiten von Information über das Verhalten des Tragsystemes Gewölbe-Bodenmaterial. Während sich die Datenverarbeitungsanlage für diese Aufgabe sehr gut eignet, wird der Ingenieur von der Verantwortung nicht entoben, das statische Modell und die zu berücksichtigenden Belastungsfälle zu wählen, sowie die Effekte der gemachten Voraussetzungen und die erhaltenen Resultate oder Informationen zu beurteilen. Alle nicht programmierbaren Entscheide — und das werden immer die wichtigen sein — können dem Ingenieur von der Maschine nicht abgenommen werden.

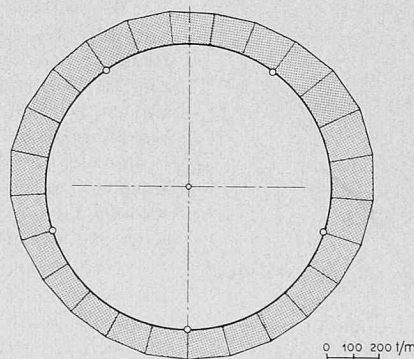
5. Zusammenfassung und Schlussfolgerung

Es sind die verschiedenen Entwicklungsstufen der Computer-Programmiersprachen dargestellt. Wesen und Zweck der in dieser Hierarchie an letzter Stelle stehenden problemorientierten Sprachen sind vom Standpunkt der Ingenieurpraxis aus beleuchtet. Die Anwendung des STRESS-Systemes wird an einem praktischen Beispiel erläutert.

Die Entwicklung der problemorientierten Sprachen ist aus dem Bedürfnis nach einem dem Ingenieur angepassten Kommunikationsniveau mit dem Computer und nach einem modular aufgebauten Programmsystem entstanden. Ihr Anwendungsgebiet liegt vor allem



a) Biegemomente



b) Normalkräfte

Bild 4. Schnittkräfte bei geneigter Geländelinie und satten Einbau

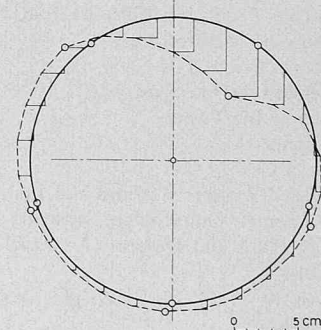


Bild 5. Deformation des Gewölbeinbaues bei geneigter Geländelinie

bei nicht im strengen Sinne repetitiven Problemen. Die Potentiale der problemorientierten und konversationsorientierten Computeranwendung werden mit der Möglichkeit von Timesharing nochmals gesteigert. Durch die Verbesserung der Mensch-Maschine-Kommunikation soll der Computer in den ganzen Prozess eines Projektes integriert werden. Damit wird der Weg zu einer sinnvollen Interaktion zwischen Mensch und Computer geebnet, bei der dem Computer nicht mehr nur die Rolle eines Rechenautomaten, sondern vermehrt die einer Maschine für die Aufnahme, Verarbeitung, Speicherung und Wiedergabe von Informationen im erweiterten Sinne zukommt. Die Aufgaben des Menschen in einem solchen System konzentrieren sich immer mehr auf Beurteilung von Situationen und das Treffen der nicht programmierbaren Entscheide. Die heutige Entwicklung ist noch nicht bei einem solch idealen System angelangt; sie bewegt sich aber deutlich in dieser Richtung [5].

Der Verfasser möchte *M. Hirt*, dipl. Ing., für seine wertvolle Mitarbeit bei der Behandlung des numerischen Beispiels und der Durchsicht des Manuskriptes bestens danken.

Literatur

- [1] ASCE Fourth Conference on Electronic Computation, Los Angeles Sept. 1966. «Journal of the Structural Division» ASCE Vol. 92, No. ST6, Dec. 1966.
- [2] *Miller C.L.*: COGO a Computer Programming System for Civil Engineering Problems, Dept. of Civ. Eng., MIT Aug. 1961.
- [3] *Fenves S.J., Logcher R.D., Mauch S.P. and Reinschmidt K.F.*: STRESS A User's Manual. MIT Press Cambridge, Mass. 1964.
- [4] *Fenves S.J., Logcher R.D. and Mauch S.P.*: STRESS Reference Manual. MIT Press Cambridge, Mass. 1964.
- [5] *Roos D.*: ICES System Design. MIT Press Cambridge, Mass. 1966.
- [6] *Pall G.A.*: Über die Anwendung problemorientierter Programmiersprachen bei der Lösung von statischen Problemen. «Der Stahlbau», Dez. 1964.
- [7] *Mauch S.P.*: STRESS eine problemorientierte Computer-Sprache für die Berechnung von Stabwerken. Schweiz. Zentralstelle für Baurationalisierung 1965.
- [8] *Fenves S.J.*: The Role of the STRESS Programming System in Structural Engineering. Presented at the International Symposium on the Use of Computers in Structural Engineering. University of New Castle, England, July 1966.
- [9] *Logcher R.D. and Sturman G.*: STRUDL — A Computer System for Structural Design. «Journal of the Structural Division», ASCE Vol. 92, No. ST6, Dec. 1966.

Adresse des Verfassers: Dr. S. P. Mauch, Waldegg, 8911 Oberlunkhofen.

Schrifttum Frank Lloyd Wright DK 72.071.1:002

In Heft 4 (S. 93) der SBZ 1966 wurden drei kleine, damals neu erschienene Schriften von Frank Lloyd Wright besprochen. Durch Vermittlung unseres Kollegen H. Elmer erhalten wir nun eine Liste aller Publikationen von Mr. und Mrs. Frank Lloyd Wright mit Angabe der jeweiligen Verlage. Diesen Überblick stellte *Richard Carney*, Sekretär der Frank Lloyd Wright Foundation, zusammen. Da gerade die Bücher von Mrs. Wright bei uns noch wenig bekannt sind, dürfte die nachfolgende Zusammenstellung einem besonderen Leserinteresse begegnen.

Frank Lloyd Wright

Horizon Press, 156 Fifth Avenue, New York City, New York:
 The Future of Architecture¹⁾
 The Natural House¹⁾
 An American Architecture
 The Story of the Tower
 A Testament
 The Living City¹⁾
 Frank Lloyd Wright: Writings and Buildings
 The Solomon R. Guggenheim Museum
 Buildings, Plans and Designs (A special facsimile of 1910 Wasmuth Portfolio)
 The Work of Frank Lloyd Wright (A special facsimile of the Wendingen edition)

¹⁾ Als Paperback editions der New American Library besprochen in SBZ 1966, H. 4, S. 93

The New American Library, 501 Madison Avenue, New York City, New York. Paperback editions of:
 The Future of Architecture
 The Natural House
 The Living City

The World Publishing Company, 2231 West 110th Street, Cleveland, Ohio. Paperback Edition of:
 Frank Lloyd Wright: Writings and Buildings

Duell, Sloan and Pearce, 124 E. 30th Street, New York City, New York: An Autobiography.
 On Architecture

Doubleday and Company, Garden City, Long Island, New York: Architecture: Man In Possession of His Earth

Olgivanna Lloyd Wright (Mrs. Frank Lloyd Wright)

Horizon Press, 156 Fifth Avenue, New York City, New York:
 Our House
 The Shining Brow—Frank Lloyd Wright
 The Roots of Life
 When Past Is Future (A boxed set of the 3 books above)
 Frank Lloyd Wright: His Life, His Work, His Words (Dec. 1966)

*

Wer den Price Co. Tower in Bartlesville von Frank Lloyd Wright mit dem Lever House in New York vergleicht (soweit, als ein Vergleich möglich ist), wird nicht erstaunt sein, von Mrs. Wright zu erfahren, dass der Meister über den vielbesprochenen Glaskasten anders dachte, als die Menge der Bewunderer.

In ihren Büchern erzählt Mrs. Olgivanna Lloyd Wright so erfrischend aus dem Leben in Taliesin und über den grossen Architekten, dass sich kaum entscheiden lässt, was faszinierender ist: diese Schilderungen oder die Entwicklung persönlicher Auffassungen durch die Autorin. Aufdrängen wird sich dem Leser jedenfalls der Eindruck, auf ungewöhnliche Kostbarkeiten gestossen zu sein. H. E.

Wettbewerbswesen

DK 72.092

Die *Sektion Zürich* des Schweizerischen Ingenieur- und Architektenvereins und die *Ortsgruppe Zürich* des Bundes Schweizer Architekten haben kürzlich gemeinsam ein Schreiben an die Gemeindeverwaltungen der Kantone Aargau, Glarus, Graubünden, Schaffhausen, Zug und Zürich gerichtet. Darin werden die Vorteile dargelegt, welche Wettbewerbe für die Projektierung eines Bauvorhabens bieten. In einem Beiblatt werden sodann einige kritische Argumente widerlegt, die gegen die Veranstaltung von Wettbewerben gelegentlich ins Feld geführt werden. Wir hoffen, dass das gut abgewogene, initiative Vorgehen der genannten Verbandsgruppen auch andernorts aufgegriffen wird. Solche Aufklärungen sind heute notwendig und haben auch Erfolg. G. R.

Schreiben des ZIA und der Ortsgruppe Zürich des BSA an die Gemeindeverwaltungen (Bau- und Schulbehörden)

Von einzelnen Baubehörden und in der Tagespresse wird von Zeit zu Zeit die Frage aufgeworfen, ob architektonische Wettbewerbe, wie sie durch die Normen des Schweiz. Ingenieur- und Architektenvereins und des Bundes Schweizer Architekten umschrieben sind, noch sinnvoll seien. Wir gestatten uns deshalb, auf einige Aspekte des architektonischen Wettbewerbes hinzuweisen.

Wer ein Bauvorhaben realisieren will, wird sich fragen, welches Vorgehen das beste Resultat erwarten lässt: Sollen Einzelaufträge erteilt werden, oder führt ein architektonischer Wettbewerb besser zum Ziel? Dass Einzelaufträge nur einen beschränkten Überblick geben können, liegt auf der Hand. Ein Wettbewerb dagegen vermittelt eine Vielfalt möglicher Varianten, gibt umfassend Auskunft über die beste Ausnutzung des Terrains und die zweckmässigste Lösung der Bauaufgabe. Die grosse Anzahl von Projekten bietet den *Vorteil der Auswahl* auf Grund von Vergleichen in organisatorischer, wirtschaftlicher und ästhetischer Hinsicht.

Erster Grundsatz bei der Beurteilung der Projekte ist die *Wahrung der Objektivität*. Der Umstand, dass die Wettbewerbsarbeiten anonym eingereicht werden, schafft hierfür die besten Voraussetzungen. Die Prämierung erfolgt ausschliesslich auf Grund der Qualität, nach wirtschaftlichen und architektonischen Gesichtspunkten. Dieses Vorgehen gibt dem Veranstalter die Gewähr, dass die beste Lösung zur Ausführung empfohlen werden kann.