

442 | Juni 1984

## SCHRIFTENREIHE SCHIFFBAU

Gerhard Jensen

**Ein Programmsystem zur  
2-dimensionalen Berechnung der  
Potentialströmung um Körper in  
unbeschränkter Flüssigkeit**

**TUHH**

*Technische Universität Hamburg-Harburg*

**Ein Programmsystem zur 2-dimensionalen Berechnung der Potentialströmung um Körper in unbeschränkter Flüssigkeit**

Gerhard Jensen, Hamburg, Technische Universität Hamburg-Harburg, 1984

© Technische Universität Hamburg-Harburg  
Schriftenreihe Schiffbau  
Schwarzenbergstraße 95c  
D-21073 Hamburg

<http://www.tuhh.de/vss>

**EIN PROGRAMMSYSTEM ZUR 2-DIMENSIONALEN BERECHNUNG DER  
POTENTIALSTRÖMUNG UM KÖRPER IN UNBESCHRÄNKTER FLÜSSIGKEIT**

**GERHARD JENSEN**

**JUNI 1984**

Inhalt

0 Einführung	2
1 Theoretische Grundlagen	3
1.1 Problemstellung	3
1.2 Lösungsansatz mit der Galerkin-Finite-Elemente-Methode	3
1.3 Zusammenhang zwischen Potentialen auf der Körperoberfläche $S$ und im Flüssigkeitsraum $R_n$	4
1.4 Beschränkung des Problems auf einen Teilraum	5
1.5 Endgültige Formulierung	5
2 Einführung der isoparametrischen Elemente	7
2.1 Auswahl der Elemente	7
2.2 Berechnung der Ableitungen	8
2.3 Berechnung der Integrale	8
3 Programmbeschreibung	9
3.1 Beschreibung der Netzgeometrie	9
3.2 Unterprogramm POT2D	9
3.3 Unterprogramm MALEN	11
3.4 Programm HYDMAS	12
3.5 Unterprogramm UMGEN	13
3.6 Unterprogramm RASTGEN	13
3.7 Unterprogramme DATWRIT, LESEN und EINDRUCK	13
4 Anwendungen	14
4.1 Das interaktive Programm INTAKTPOT	14
4.2 Anwendungen von HYDMAS	18
4.3 Berechnung der Strömung um einen Kreiszyylinder mit verschiedenen Netzen	19
Literatur	21
Anhang 1, Bilder	22
Anhang 2, Programmtexte	25

## 0. Einführung

Das hier beschriebene Programmsystem POT2D ermöglicht die 2-dimensionale Berechnung der zirkulationsfreien Umströmung beliebiger Körper in unbeschränkter idealer Flüssigkeit. Die Berechnung erfolgt mit einer Finite-Elemente-Methode nach Jami-Lenoir/1/. Neben der Berechnung des Potentials gibt es Routinen zur Berechnung und zum Plotten der Geschwindigkeitsfelder und zur Berechnung von hydrodynamischen Massenmatrizen. Es gibt zwei Generierungs- und Digitalisierungsmöglichkeiten.

Im ersten Abschnitt wird eine Herleitung der Gleichungen angegeben, wobei ein von den Darstellungen in /1/ und /2/ etwas abweichender Weg gezeigt wird.

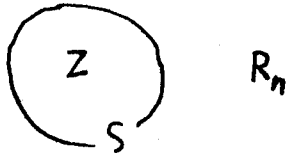
Im zweiten Abschnitt wird die Einführung der isoparametrischen Elemente beschrieben, und im dritten Abschnitt wird die Struktur der Fortranprogramme und die Funktion der wichtigsten Routinen erläutert. Dies soll dem Anwender ermöglichen Anpassungen an bestimmte Zwecke selbst vorzunehmen.

Im letzten Abschnitt sind schließlich einige Rechenbeispiele und Ergebnisse angegeben.

## 1. Theoretische Grundlagen

### 1.1 Problemstellung

Es sei  $R_n$  ein unbeschränkter  $n$ -dimensionaler euklidischer Raum, aus dem der Bereich  $Z$  ausgespart ist.  $S$  sei der gemeinsame Rand von  $Z$  und  $R_n$ .



Gesucht ist die reelle Funktion  $f$  (das Potential) in  $R_n$ , die folgende Bedingungen erfüllt:

$$\text{Laplacegleichung: } \Delta f = 0 \text{ in } R_n \quad (1)$$

$$\text{Randbedingung: } \frac{\partial f}{\partial \vec{n}} = g \text{ auf } S, \quad (2)$$

wobei  $g$  eine gegebene reelle Funktion auf  $S$  ist.  $\vec{n}$  ist ein in den Körper hineinzeigender Normalenvektor.

$$\lim_{\text{Abstand vom Körper} \rightarrow \infty} \Delta f = 0$$

Das Problem tritt z.B. bei der zirkulationsfreien Strömung einer idealen Flüssigkeit um einen schwingenden Körper  $Z$  auf;  $g$  ist dabei die Geschwindigkeitskomponente des Körpers normal zur Körperoberfläche  $S$ ;  $f$  ist das Potential der Strömung. Vorausgesetzt ist dabei, daß die Flüssigkeit unendlich weit vom Körper ruht.

Das Problem tritt auch bei der Störung einer Parallelströmung durch einen ruhenden Körper auf, wenn wie vorher die Flüssigkeit inkompressibel und die Strömung zirkulationsfrei ist. In diesem Fall ist  $f$  das Potential der Abweichung zwischen wirklicher Strömung und Parallelströmung;  $g$  ist die Normalkomponente der Anströmgeschwindigkeit auf der Körperoberfläche  $S$  und in unendlicher Entfernung vom Körper soll die Parallelströmung ungestört sein.

### 1.2 Lösungsansatz mit der Galerkin-Finite-Elemente-Methode

Zur Lösung dieser Differentialgleichung soll hier die Galerkin-Finite-Elemente-Methode /4/ benutzt werden:

$$f(P) = \sum_i f_i w_i(P) \text{ sei eine Näherungslösung.} \quad (3)$$

Die Ansatzfunktionen  $w_i$  sind linear unabhängige, stetige und stückweise differenzierbare Funktionen des Ortes  $(P)$ , die nur in den Elementen (finite, zusammenhängende, nicht überlappende Teilbereiche von  $R_n$ ) von Null verschieden sind, zu denen der Knoten  $i$  gehört. Am Knoten  $i$  haben sie den Wert 1, an allen anderen Knoten den Wert 0. Damit haben sie insbesondere an allen Elementrändern, zu denen der Knoten  $i$  nicht gehört, den Wert 0.  $f_i$  sind daher die Werte des Potentials am Knoten  $i$ . Es wird, statt die Laplacegleichung exakt zu erfüllen, nur die Erfüllung im gewichteten Mittel mit verschiedenen Gewichtsfunktionen  $w_j$  gefordert:

$$\int_{R_n} \nabla f w_j dR_n = 0 \quad (4)$$

Hierbei werden als Gewichtsfunktionen die Ansatzfunktionen benutzt. Damit erhält man ein lineares Gleichungssystem für die  $f_i$ .

In dieser Form ist diese Formulierung jedoch nicht anwendbar, weil der Flüssigkeitsraum  $R_n$  unendlich ausgedehnt ist, so daß man unendlich viele Elemente anordnen müßte. Außerdem wird, anders als bei der klassischen Galerkin Methode, die Randbedingung von den Ansatzfunktionen nicht erfüllt.

### 1.3 Zusammenhang zwischen dem Potential auf der Körperoberfläche $S$ und im Flüssigkeitsraum $R_n$

Es sei:

$f(P)$  das Potential an einer Stelle  $P$  (gesucht)

$G(M,P)$  das Potential (bekannt) an der Stelle  $P$  in Folge einer Singularität (Quelle) an der Stelle  $M$ .

$G$  heißt Greenfunktion mit folgenden Eigenschaften:

$$\Delta G = \delta(M,P) \Rightarrow \int_{R_p} \Delta G(M,P) dR_p = 1, \text{ wenn } M \text{ im Inneren der Umgebung } R_p \text{ von } P \text{ liegt. Dies wird im folgenden vorausgesetzt.}$$

=>

$$\int_{R_n} f(P) \Delta G(M,P) dR_n = f(M)$$

Hierauf läßt sich der zweite Greensche Satz /3/ anwenden:

$$\int_{R_n} f \Delta G - G \Delta f dR_n = \int_S \left( f \frac{\partial G}{\partial \vec{n}} - G \frac{\partial f}{\partial \vec{n}} \right) dS_p$$

Mit  $\Delta f = 0$  in  $R_n$

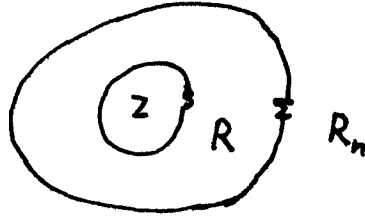
$$\Rightarrow f(M) = \int_S \left( f \frac{\partial G}{\partial \vec{n}} - G \frac{\partial f}{\partial \vec{n}} \right) dS_p$$

Mit  $\frac{\partial f}{\partial \vec{n}} = g$  auf  $S$  und dem Summenansatz für  $f$  erhält man:

$$f(M) = \sum_i f_i \int_S w_i(P) \frac{\partial G}{\partial \vec{n}}(M,P) dS_p - \int_S g(P) G(M,P) dS_p \quad (6)$$

$i=1..n$  sind die Knoten auf dem Körperrand.

## 1.4 Beschränkung des Problems auf einen Teilraum



Umschließt man den Körper nun mit einer Kontrollfläche  $\Sigma$  und nennt jetzt nur den eingeschlossenen Flüssigkeitsraum  $R$ , so kann man den ersten Greenschen Satz /3/

$$\int_R w \Delta f \, dR = \int_S w \frac{\partial f}{\partial n} \, dS - \int_R \nabla w \nabla f \, dR$$

auf (4) anwenden und (2) einsetzen:

$$0 = \int_S w_j g(P) \, dS + \int_{\Sigma} w_j \frac{\partial f}{\partial n} \, d\Sigma - \int_R \nabla w_j \nabla f \, dR \quad (7)$$

Setzt man hierin wieder den Summensatz (3) für  $f$  ein, so erhält man folgendes Gleichungssystem für die Knotenpotentiale  $f_i$ :

$$0 = \int_S w_j g(P) \, dS + \sum_i f_i \int_{\Sigma} w_j \frac{\partial w_i}{\partial n}(P) \, d\Sigma - \sum_i f_i \int_R \nabla w_j \nabla w_i \, dR \quad (8)$$

Die Gleichung soll nur für die  $j=1..N$  Knoten innerhalb von  $\Sigma$  angewendet werden. Dann ist das  $w_j$  im zweiten Summanden = 0, und dieser kann wegfallen. Im dritten Summanden treten die unbekannt Potentiale der Außenrandknoten auf, so daß mehr Unbekannte als Gleichungen vorhanden sind. Deshalb wird (6) für die unbekannt Potentiale der Außenrandknoten eingesetzt:

$$0 = \int_S w_j g(P) \, dS - \sum_{i=1}^N f_i \int_R \nabla w_j \nabla w_i \, dR \quad (9)$$

$$- \sum_M \left( \sum_{i=1}^n f_i \int_S w_i(P) \frac{\partial G}{\partial n}(M,P) \, dS - \int_S g(P) G(M,P) \, dS \right)$$

$$\int_R \nabla w_M \nabla w_j \, dR$$

Hierin bezeichnet  $M$  Knoten auf dem Außenrand  $\Sigma$ .

## 1.5 Endgültige Formulierung

In Gleichung (9) kann die Summation über die Knoten auf dem Körpertrand ( $i=1..n$ ) auch über alle Knoten ( $i=1..N$ ) durchgeführt werden, da die Summanden für Knoten, die nicht auf  $S$  liegen ohnehin 0 werden. Vertauscht man nun im 3. Summanden die Summationen über  $M$  und  $i$  und bringt die Terme, die  $f_i$  enthalten, auf die linke Seite, so erhält man ein lösbares lineares Gleichungssystem:

$$\sum_{i=1}^N K_{ij} f_i = R_j \quad (10)$$

mit

$$K_{ij} = \int_R \nabla w_j \nabla w_i \, dR + \sum_M \left( \int_S w_i(P) \frac{\partial G}{\partial \bar{M}}(M,P) \, dS \right) \int_R \nabla w_M \nabla w_j \, dR$$

$$R_j = \int_S w_j g(P) \, dS + \sum_M \left( \int_S g(P) G(M,P) \, dS \right) \int_R \nabla w_M \nabla w_j \, dR$$

## 2. Einführung der isoparametrischen Elemente

### 2.1 Auswahl der Elemente

Wie bei allen Galerkin Methoden hängt die erzielte Genauigkeit neben der Anzahl der Ansatzfunktionen (hier Anzahl der Knoten) wesentlich von der Anpassbarkeit der Ansatzfunktionen an die exakte Lösung ab. Da sich bei Finiten Elementen lokale Polynome wegen ihrer einfachen Berechenbarkeit anbieten, sollten hier, um auch bei grober Diskretisierung gute Anpassung an den vorhandenen Verlauf zu erhalten, Polynome 2. Grades möglich sein.

Auch für die Beschreibung eines krummlinigen Körperperrandes sollten zur Koordinateninterpolation mindestens quadratische Ansätze gemacht werden.

Es bot sich daher an isoparametrische Elemente /5/ zu verwenden. Es werden viereckige isoparametrische Elemente benutzt, an deren Rand sich 4..8 Knoten befinden. (Die Knoten 4..8 müssen nicht vorhanden sein) Es wird ein lokales Koordinatensystem  $r,s$  benutzt in dem die Ränder bei  $r_{i=+-1}$  und die seitlichen Zwischenknoten auf den Koordinatenachsen liegen.

Ansatzfunktionen in lokalen Koordinaten:

$$H_1 = 0,25(1+r)(1+s) - 0,5H_5 - 0,5H_8$$

$$H_2 = 0,25(1-r)(1+s) - 0,5H_5 - 0,5H_6$$

$$H_3 = 0,25(1-r)(1-s) - 0,5H_6 - 0,5H_7$$

$$H_4 = 0,25(1+r)(1-s) - 0,5H_7 - 0,5H_8$$

$$H_5 = 0,5(1-r^2)(1+s) \text{ , wenn Knoten 5 vorhanden}$$

$$H_6 = 0,5(1-s^2)(1-r) \text{ , wenn Knoten 6 vorhanden}$$

$$H_7 = 0,5(1-r^2)(1-s) \text{ , wenn Knoten 7 vorhanden}$$

$$H_8 = 0,5(1-s^2)(1+r) \text{ , wenn Knoten 8 vorhanden}$$

$$H_i = 0 \text{ , wenn Knoten } i \text{ nicht vorhanden, für } i = 5,8$$

Die Ansatzfunktionen  $w$  setzen sich nun aus den lokalen Ansatzfunktionen  $H$  auf allen Elementen, zu denen der Knoten gehoert, zusammen.

Die Koordinaten ergeben sich aus lokalen Koordinaten und den Knotenkoordinaten  $X_i, Y_i$ :

$$x = \sum_{i=1}^8 H_i X_i$$

$$y = \sum_{i=1}^8 H_i Y_i$$

Auf die gleiche Weise wird das Potential interpoliert:

$$f = \sum_{i=1}^8 H_i f_i$$

Für die Ableitungen nach  $r$  und  $s$  gilt entsprechendes.

## 2.2 Berechnung der Ableitungen

Der Zusammenhang zwischen den Ableitungen nach  $x, y$  und denen  $r, s$  ist durch die Jacobische Matrix  $J$  gegeben:

$$\begin{bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

Damit lassen sich die notwendigen Ableitungen zur Bestimmung der Gradienten der Ansatzfunktionen und der Normalenvektoren bestimmen.

## 2.3 Berechnung der Integrale

Alle Integrationen werden elementweise numerisch in den lokalen Koordinaten durchgeführt. Damit ergeben sich stets die festen Integrationsgrenzen  $-1, +1$ .

### 2.3.1 Berechnung der Integrale über die Körperoberfläche

Die Linienintegrale  $dS$  um den Körper können aus den Integralen der entsprechenden Elementränder zusammengesetzt werden. Wird z.B. entlang des Randes  $r=1$  integriert, so ist das Linienelement:

$$dS = \left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2 ds$$

Entsprechendes gilt auch für die anderen Ränder.

### 2.3.2 Berechnung der Integrale über den Berechnungsraum

Die Flächenintegrale  $dR$  über den Berechnungsraum werden aus den Integralen über die einzelnen Elemente zusammengesetzt. Das Flächenelement ist nach [3]:

$$dR = \sqrt{E G - F^2} dr ds$$

mit

$$E = \left(\frac{\partial x}{\partial r}\right)^2 + \left(\frac{\partial y}{\partial r}\right)^2$$

$$F = \frac{\partial x}{\partial r} \frac{\partial y}{\partial s} + \frac{\partial y}{\partial r} \frac{\partial x}{\partial s}$$

$$G = \left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2$$

### 3. Programmbeschreibung

Alle Programme sind in FORTRAN77 geschrieben und laufen auf der VAX11/780 des Institut für Schiffbau. Außer bei den interaktiven Programmteilen und bei den OPEN-Statements wurde versucht keine Abweichungen vom Standard zu verwenden.

#### 3.1 Beschreibung der Netzgeometrie

Die Beschreibung der Netzgeometrie ist für alle Programme gleich. Deshalb soll sie zuerst einmal angegeben werden. Für die Generierungsprogramme ist sie Ergebnis und für die Berechnung des Potentials, der hydrodynamischen Massen und der Geschwindigkeitsfelder ist sie Eingangsgröße. Das Netz bestehe aus M Elementen mit N Knoten. Die Knoten werden von 1..N nummeriert. Dabei ist zu beachten, daß die Knoten auf dem Außenrand  $\Sigma$  die größten Knotennummern haben. Zu jedem Knoten befinden sich im Feld PK seine Koordinaten: PK(1,I) = X-Koordinate des I-ten Knoten, PK(2,I) = Y-Koordinate des I-ten Knoten.

Daneben gibt es ein Feld BODKN dessen erste K Elemente die Nummern der K Knoten auf der Körperoberfläche S sind. Das Feld SIGKN enthält auf seinen ersten L Plätzen die Nummern der L Knoten auf dem Außenrand  $\Sigma$ . Dazu gibt es ein Feld ZEIGER, das zu jedem Knoten einen Zeiger mit folgenden Eigenschaften enthält:  
 ZEIGER(I)>0, wenn sich der Knoten I auf dem Körperperrand S befindet und I=BODKN(ZEIGER(I))  
 ZEIGER(I)<0, wenn sich der Knoten I auf dem Außenrand  $\Sigma$  befindet und I=SIGKN(-ZEIGER(I))  
 ZEIGER(I)=0 für alle anderen Knoten.

Die Zugehörigkeit der Knoten zu den Elementen wird in Feld KN gespeichert. KN(I,J)= Nummer des I-ten Knoten auf dem J-ten Element (J=1..M). Die Nummerierung erfolgt gemäß Abbildung in 2.1. Ist der I-te Knoten bei einem Element nicht vorhanden, so ist KN(I,J)=0. Liegt das Element am Körperperrand so muß die Nummerierung so erfolgen, daß die Seite mit dem 3. und 4. Knoten den Körperperrand bildet. Befindet sich das Element in einer Körperecke, so daß zwei Kanten auf S liegen, so sollen dies die Kanten mit den Knoten 3 und 4 sowie mit 4 und 1 sein. Andere Kanten können nicht Körperoberfläche sein.

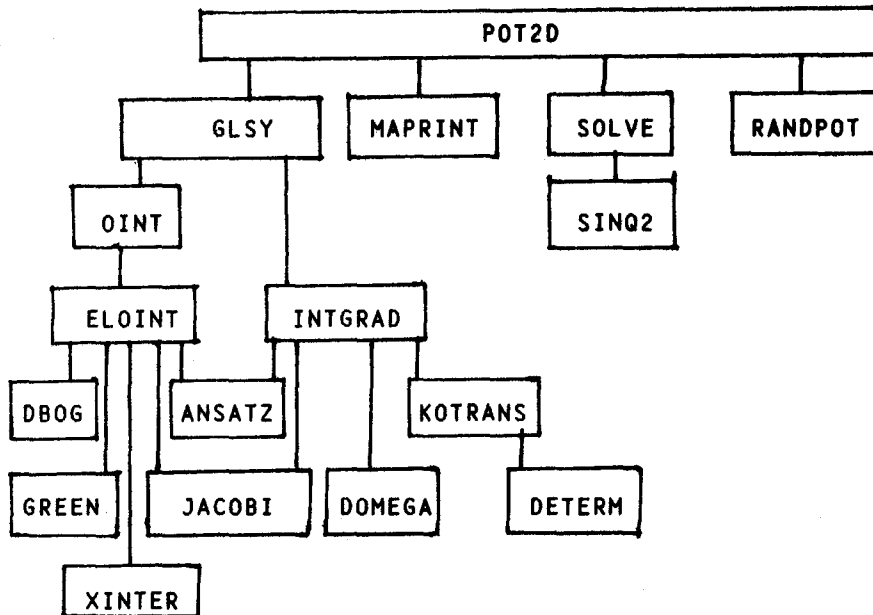
Im Feld GV ist bei GV(1,i) die X-Komponente und bei GV(2,i) die Y-Komponente der Geschwindigkeit des Körperknotens BODKN(i) abgelegt.

#### 3.2 Unterprogramm POT2D

POT2D berechnet mit seinen Hilfsroutinen die Knotenpotentiale für ein gegebenes Netz und Randbedingungen. Die Bedeutung der Aufrufparameter kann man den Kommentaren am Programmanfang entnehmen.

### 3.2.1 Struktur

Hier die wichtigsten Unterprogrammaufrufe:



### 3.2.2 Berechnung der Koeffizienten

Die Berechnung der Koeffizienten  $KIJ(i,j)$  und der rechten Seiten  $RI(j)$  des Gleichungssystems erfolgt im Unterprogramm GLSY. Dazu werden zunächst im Unterprogramm OINT alle Integrale über die Körperoberfläche gebildet. Nach der Ausführung von OINT ist:

$$RI(j) = \int w_j g \, dS, \text{ für alle } j \text{ Knoten auf } S$$

$$GG(-ZEIGER(M)) = \int g(P) G(M,P) \, dS_p, \text{ für alle } M \text{ Knoten auf } \Sigma$$

$$IWDG(-ZEIGER(M), ZEIGER(i)) = \int w_i(P) \frac{\partial G}{\partial n}(M,P) \, dS$$

Dann wird für alle Elemente L  $IGG(i,j) = \int w_i w_j \, dR_L$  berechnet.

Zu  $KIJ(KN(i,L), KN(j,L))$  wird  $IGG(i,j)$  addiert.

Für alle Knoten M des Elements, die am Außenrand liegen, wird aus IWDG und IGG der zweite Term von  $KIJ$  in (10) berechnet und addiert und aus IGG und GG der zweite Term von  $RI$  berechnet und addiert.

### 3.2.3 Ausgeben des Gleichungssystems

Mit dem Unterprogramm MAPRINT können auf Wunsch die Koeffizientenmatrix und die rechten Seiten ausgegeben werden.

### 3.2.4 Lösen des Gleichungssystems

In der Routine SOLVE wird das Gleichungssystem für die nicht auf dem Außenrand  $\Sigma$  befindlichen Knoten mittels des Unterprogramms SIMQ2 gelöst. Die Lösung steht nach Ausführung des Unterprogramms im Feld RI. Sollte das Gleichungssystem singular sein, wird eine Fehlermeldung auf Datei 6 ausgegeben und die Programmausführung abge-

brochen. Dies sollte aber bei richtiger Eingabe der Geometrie nicht vorkommen.

### 3.2.5 Berechnung der Potentiale an den Außenrandknoten

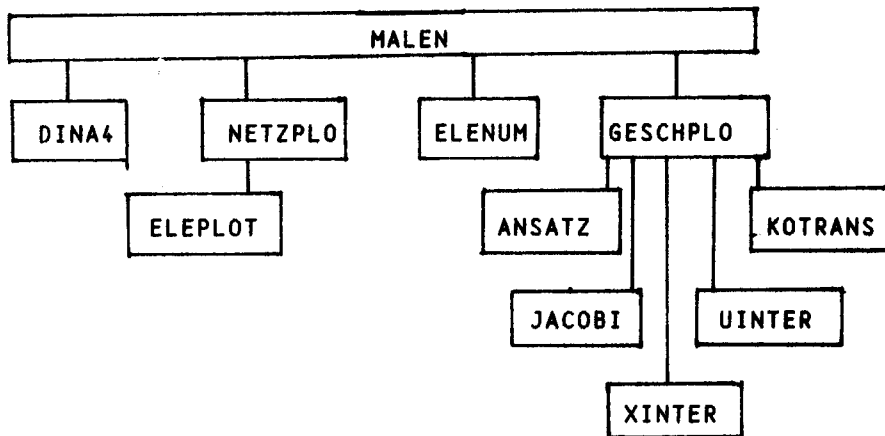
Für einige weitere Berechnungen kann es nützlich sein, wenn auch die Potentiale an den Knoten auf  $\Sigma$  bekannt sind. Deshalb werden im Unterprogramm RANDPOT nach Gleichung (6) ihre Potentiale aus dem jetzt bekannten Potential an der Körperoberfläche und IWDG (siehe 3.2.2) berechnet.

## 3.3 Unterprogramm MALEN

Das Unterprogramm MALEN dient dem Plotten des Berechnungsnetzes und des Geschwindigkeitsfeldes. Es werden Calcomp Plotroutinen benutzt. Einzelheiten zum Aufruf von MALEN kann man den Kommentaren im Unterprogramm entnehmen.

### 3.3.1 Programmstruktur

Hier die wichtigsten Unterprogrammaufrufe



### 3.3.2 Maßstab

Das Unterprogramm DINA4 berechnet einen Maßstabsfaktor SCAL so, daß das Bild des Netzes gerade noch auf eine DIN A4 Seite paßt (mit Rand).

Der beim Aufruf von MALEN anzugebende Parameter GLCM bestimmt die Länge der Geschwindigkeitspfeile. GLCM ist die Länge eines Geschwindigkeitsvektors vom Betrag 1 in Zentimetern.

### 3.3.3 Netzausgabe

Das Unterprogramm NETZPLO plottet alle Elementgrenzen und nummeriert die Knoten. Falls beim Aufruf von MALEN angegeben werden mit ELENUM die Nummern der Elemente an den Schnittpunkt der Diagonalen der Elemente geschrieben.

### 3.3.4 Berechnen und Plotten der Geschwindigkeitspfeile

Im Unterprogramm GESCHPLO werden an 9 Stellen in jedem Element, insbesondere an 3 Stellen am Körperperrand, falls das Element an der Körperoberfläche liegt, die Geschwindigkeitsvektoren  $[u,v]$  aus  $u = \partial\phi/\partial x$  und  $v = \partial\phi/\partial y$  berechnet und ein entsprechender Geschwindigkeitspfeil in das Netz geplottet.

Falls das Störpotential für eine stationäre Anströmung berechnet wurde, kann diese

Anströmgeschwindigkeit zu den Störgeschwindigkeiten addiert werden, so daß das Bild der tatsächlichen Umströmung entsteht.

### 3.4 Programm HYDMAS

Zur Bestimmung der hydrodynamischen Masse für sinusförmige periodische Bewegungen ohne Berücksichtigung einer freien Oberfläche kann das Programm HYDMAS benutzt werden.

#### 3.4.1 Berechnungsmethode

Nach /6/ werden die Elemente der hydrodynamischen Massenmatrix  $M$  berechnet zu:

$$M_{kj} = 1/\omega^2 \int p_k \bar{n} \bar{q}_j \, ds$$

Darin ist  $\omega$  die Kreisfrequenz der Schwingung,  $\bar{q}_j$  ein Verschiebungsansatz für den Körper (Freiheitsgrad) und  $p_k$  ist der mit  $\omega$  periodisch schwankende Druckanteil auf die Körperoberfläche infolge einer Bewegung des Körpers in Richtung des Freiheitsgrades  $k$  (Verschiebungsansatz  $\bar{q}_k$ ).  $\bar{n}$  ist der Normalenvektor auf die Oberfläche. Unter Vernachlässigung des hydrostatischen Drucks ergibt sich der mit  $\omega$  periodisch schwankende Druckanteil nach der Bernoulli-Gleichung zu

$$p_k = \rho f(\dot{q}_k) \quad \cdot \text{ bedeutet Ableitung nach der Zeit}$$

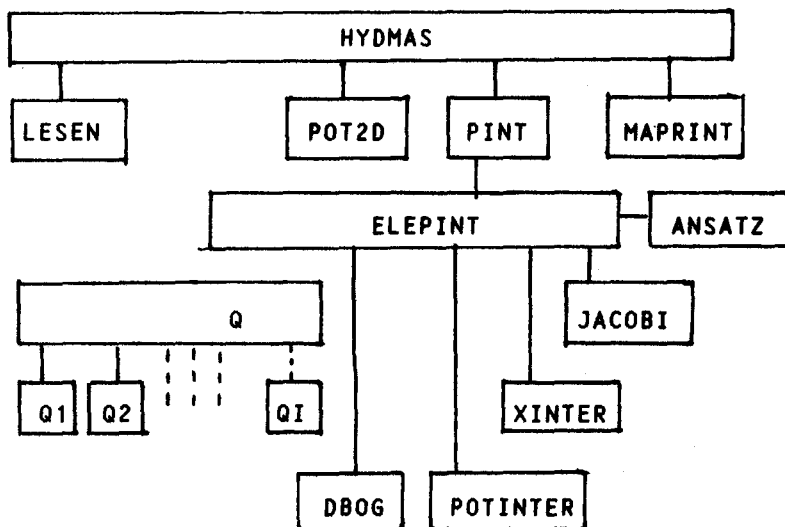
$$\Rightarrow p_k = \rho \omega^2 f(q_k) \quad \rho \text{ ist die Dichte der Flüssigkeit}$$

$f(q_k)$  ist das Geschwindigkeitspotential, wenn man  $\bar{q}_k$  als Geschwindigkeit an der Körperoberfläche einsetzt. Setzt man dies in den Ausdruck für  $M_{kj}$  ein so sieht man, daß die Abhängigkeit von der Frequenz herausfällt:

$$M_{kj} = \rho \int f(q_k) \bar{n} \bar{q}_j \, ds$$

In HYDMAS wird dieser Ausdruck dividiert durch die Dichte der Flüssigkeit und bezogen auf die Breite (2-dimensionale Berechnung) berechnet.

#### 3.4.2 Programmstruktur



Hierin dient das Unterprogramm LESEN zum Lesen der Daten, POT2D zur Berechnung des Potentials, PINT berechnet elementweise mit ELEPINT die Oberflächenintegrale und mit MAPRINT wird als Ergebnis die Massenmatrix gedruckt.

### 3.4.3 Beschreibung der Ansatzfunktionen

Zur Beschreibung der Verschiebungsansätze müssen ebenso viele Unterprogramme  $QI(RES,X)$  zum Problem passend geschrieben, übersetzt und angebunden werden.  $X$  ist ein Feld mit zwei Elementen, daß die  $X$ - und die  $Y$ -Koordinate eines Ortes enthält. Die  $QI$  berechnen die beiden Komponenten des Verschiebungsvektors an diesem Ort,  $RES(1),RES(2)$ . Das Programm  $HYDMAS$  und das Unterprogramm  $Q$  müssen an die Anzahl der vorhandenen Freiheitsgrade angepasst werden. Hierauf wird in Kommentaren im Programmtext hingewiesen. Auch die im nächsten Kapitel angegebenen Beispiele mögen als Erläuterung dienen.

### 3.5 Unterprogramm UMGEN, Digitalisieren der Körperoberfläche

Das Unterprogramm  $UMGEN$  generiert aus den Koordinaten der Knoten an der Körperoberfläche ein Berechnungsnetz, wie es für  $POT2D$  und  $HYDMAS$  benötigt wird (siehe 3.1), bestehend aus einer Schicht von Elementen um den Körper herum. Dabei kann durch Angabe von Parametern in der Eingabedatei bestimmt werden, ob die Elemente seitlich, außen und auf dem Körper Zwischenknoten erhalten. Eine genaue Beschreibung der Eingabedatei wird in den Kommentaren im Unterprogramm gegeben.

$UMGEN$  kann sinnvoll gemeinsam mit dem Digitalisieretisch eingesetzt werden. Es werden nur im Gegenuhrzeigersinn nacheinander die Knoten an der Körperoberfläche sowie ein  $POI$  zur Beschreibung etwaiger Drehbewegungen des Körpers digitalisiert. Aus diesen Daten kann mit dem Programm  $DIGEN$  interaktiv eine Eingabedatei für  $UMGEN$  erzeugt werden.

### 3.6 Unterprogramm RASTGEN

Eine andere Hilfe zur Erzeugung von Daten für  $POT2D$  und  $HYDMAS$  bietet das Unterprogramm  $RASTGEN$ . Als Eingabe wird ein Netz isoparametrischer Elemente sowie Listen der Knoten die sich an der Körperoberfläche befinden und der Knoten auf dem Außenrand benötigt. Dummyknoten erlauben eine die automatische oder manuelle Erzeugung oder Ergänzung des isoparametrischen Netzes vereinfachende Knotennummerierung.

Eine genaue Beschreibung der erforderlichen Eingabedatei wird in den Kommentaren am Beginn des Unterprogramms gegeben. Das Eingabernetz kann zur Kontrolle geplottet werden, wobei die als Körper-, Außen- oder Dummyknoten definierten Punkte besonders gekennzeichnet werden. Im nächsten Kapitel ist auch für die Anwendung von  $RASTGEN$  ein Beispiel angegeben.

### 3.7 Unterprogramme DATWRIT, LESEN und EINDRUCK

Das Unterprogramm  $DATWRIT$  schreibt alle zur Beschreibung des Netzes erforderlichen Daten in eine Datei. Von dort können sie mit dem Unterprogramm  $LESEN$  wieder eingelesen werden. Auf diese Weise können die einmal generierten Daten abgelegt werden und von anderen Programmen zur weiteren Berechnung wieder eingelesen werden. Das Programm  $Hydmas$  liest z.B. mit  $LESEN$  seine Daten ein.  $EINDRUCK$  schreibt die wesentlichen Daten, die für den Aufruf von  $POT2D$  benötigt werden formatiert in eine Tabelle. Dies soll dem Benutzer eine weitere Kontrollmöglichkeit für seine Daten geben.

## 4 Anwendungen

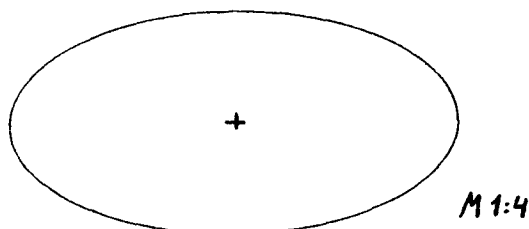
### 4.1 Das interaktive Programm INTAKTPOT

Das Generieren von Daten und die Steuerung der Berechnung und Ausgabe unterstützt das auf der VAX unter VMS lauffähige Programm INTAKTPOT. Eine Programmliste ist im Anhang 2 enthalten.

Der Programmablauf wird durch Antworten des Benutzers auf Fragen des Programms gesteuert. Daher gibt es für den Verlauf der Sitzung viele Möglichkeiten. Hier sollen nur zwei Beispiele gezeigt werden.

#### 4.1.1 Umtströmung einer Ellipse

Mit dem Digitalisiertisch seien die Koordinaten von Knoten auf der Ellipse in nachste-



hender Zeichnung in die Datei ELLIPS.DAT gebracht worden.

Inhalt der Datei (Kommentare nicht enthalten):

33			(Anzahl der Knoten)
1	28560.00	4375.000	(Koordinaten in 1/400 cm
2	28520.00	5177.000	vom Korrdinatenursprung
3	28397.00	5960.000	des Tisches gemessen)
4	28199.00	6572.000	
5	27979.00	7077.000	
	*		
	*		
32	28494.00	3591.000	
33	26729.00	4395.000	(Rotationspol)

Mit dieser Eingabedatei nun ein Beispiel für eine interaktive Sitzung um ein Plotbild der Geschwindigkeitspfeile mit einer stationären Anströmung in Richtung der X-Achse zu erhalten. Die Antworten des Benutzers sind zur Verdeutlichung unterstrichen dargestellt.

Programm zur Berechnung von Potentialstroemungen

Sollen die Netzdaten aus digitalisierten Daten der Knoten an der Koerperoberflaechе erzeugt werden? Ja/Ne JA  
 Dieses Unterprogramm erzeugt aus den mit "DIGIP" digitalisierten Daten Eingabedaten fuer die Netzgenerierung

Auf welcher Datei sind die Daten? ELLIPS.DAT

Sollen die Elemente auf der Koerperoberflaechе

Zwischenknoten haben? Ja/Ne JA

Sollen seitliche Zwischenknoten generiert werden? Ja/Ne JA

Sollen aeuessere Zwischenknoten generiert werden? Ja/Ne JA

Im folgenden brauchen nur Zahlen eingegeben zu werden wo Abweichungen von den angegebenen gewünscht werden, sonst <RET>

X-Koordinate der Rotationsachse=-4.577? <RET>

Y-Koordinate der Rotationsachse= 0.050? <RET>

Winkelgeschwindigkeit (Bogenmass)= 0.000? <RET>

X-Komponente der Anstroemgeschwindigkeit= 0.000? <RET>

Y-Komponente der Anstroemgeschwindigkeit= 0.000? 2

Sollen die Netzdaten gespeichert werden? Ja/Ne JA

Dateiname? NETZ.DAT

Fuer das Lesen und Generieren

verbrauchte CPU-Zeit: 0.860SEC

Erfolgt die Ausgabe auf den Bildschirm? Ja/Ne JA

Sollen die Eingabedaten ausgegeben werden? Ja/Ne JA

ANZAHL DER KNOTEN:				80 ANZAHL DER ELEMENTE:			16	
ELNR	LKNR	GKNR	ZEIGER	X	Y	VX	VY	
1	1	64	-1	-1.42308	-0.45000			
1	2	49	-3	-0.95524	-3.11539			
1	3	3	3	0.75000	-1.35457	0.00000	1.00000	
1	4	1	1	0.00000	0.00000	0.00000	1.00000	
1	5	65	-2	-1.33870	-1.69635			
1	6	33	0	-0.10262	-2.23498			
1	7	2	2	0.23594	-0.63144	0.00000	1.00000	
1	8	48	0	-0.71154	-0.22500			
2	1	49	-3	-0.95524	-3.11539			
2	2	50	-5	1.43747	-6.41417			
2	3	5	5	2.93469	-2.84207	0.00000	1.00000	
2	4	3	3	0.75000	-1.35457	0.00000	1.00000	
2	5	66	-4	-0.02098	-4.78239			
2	6	34	0	2.18608	-4.62812			
2	7	4	4	1.63177	-2.11607	0.00000	1.00000	
2	8	33	0	-0.10262	-2.23498			
3	1	50	-5	1.43747	-6.41417			
3	2	51	-7	6.09879	-8.79782			
			*					
			*					
			*					
15	7	30	30	0.23594	1.64866	0.00000	1.00000	
15	8	46	0	-0.10262	3.25220			
16	1	63	-31	-1.42308	1.46722			
16	2	64	-1	-1.42308	-0.45000			
16	3	1	1	0.00000	0.00000	0.00000	1.00000	
16	4	31	31	0.00000	1.01722	0.00000	1.00000	
16	5	80	-32	-1.48156	0.50861			
16	6	48	0	-0.71154	-0.22500			
16	7	32	32	-0.05767	0.50861	0.00000	1.00000	
16	8	47	0	-0.71154	1.24222			

Fuer das Schreiben der Eingabedaten

verbrauchte CPU-Zeit: 0.700SEC

Soll das Gleichungssystem ausgegeben werden? Ja/Ne NE

FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSSYSTEMS

VERBRAUCHTE CPU-ZEIT: 2.440SEC

FUER DAS LOESEN DES GLEICHUNGSSYSTEMS

VERBRAUCHTE CPU-ZEIT: 0.630SEC

Sollen die Potentiale ausgegeben werden? Ja/Ne JA

Falls eine andere Ausgabedatei bitte Dateiname eingeben,

```

sonst <RET> <RET>
  1      1.00455236   2      2.29809999   3      3.70699239
  4      5.24995184   5      6.69622135   6      7.90750504
  7      8.71909237   8      8.99596024   9      8.71909142
 10      7.90750551  11      6.69621658  12      5.24994135
 13      3.70698214  14      2.29809117  15      1.00454462

```

```

*
*
*

```

```

70      3.81306195  71      1.98265064  72      -0.00000132
73     -1.98265362  74     -3.81306338  75     -5.08447742
76     -5.52805328  77     -5.08448172  78     -3.81306767
79     -1.98265958  80     -0.00000196

```

Soll das Netz und das Geschwindigkeitsfeld geplottet werden?  
Ja/Ne JA  
Nummer der Plotdatei? 10  
Fuer das Erstellen der Plotdatei  
verbrauchte CPU-Zeit: 4.920SEC  
CPU-Zeit gesamt: 9.740 SEC  
FORTRAN STOP

Das Plotbild ist Bild 1 im Anhang 1 dargestellt dargestellt.

Die Netzdaten wurden in die Datei NETZ.DAT geschrieben. Von dort können sie für eine weitere Rechnung noch einmal gelesen werden.

Hierzu gleich ein Beispiel:

Die Ellipse soll um ihren Mittelpunkt rotieren und man will wieder ein Bild der Umströmung plotten.

Programm zur Berechnung von Potentialstroemungen

```

Sollen die Netzdaten aus digitalisierten Daten Der Knoten an
der Koerperoberflaeche erzeugt werden? Ja/Ne NE
Name der Eingabedatei? NETZ.DAT
Soll das Netz aus einem Raster generiert werden? Ja/Ne NE
Sind die Eingabedaten bereits vollstaendig vorhanden? Ja/Ne JA
Im folgenden brauchen nur Zahlen eingegeben zu werden wo
Abweichungen von den angegebenen gewuenscht werden, sonst <RET>
X-Koordinate der Rotationsachse=-4.577? <RET>
Y-Koordinate der Rotationsachse= 0.050? <RET>
Winkelgeschwindigkeit (Bogenmass)= 0.000? .5
X-Komponente der Anstroemgeschwindigkeit= 0.000? <RET>
Y-Komponente der Anstroemgeschwindigkeit= 0.000? <RET>
Sollen die Netzdaten gespeichert werden? Ja/Ne NE
Fuer das Lesen und Generieren
verbrauchte CPU-Zeit: 0.600SEC
Erfolgt die Ausgabe auf den Bildschirm? Ja/Ne JA
Sollen die Eingabedaten ausgegeben werden? Ja/Ne NE
Soll das Gleichungssystem ausgegeben werden? Ja/Ne NE
FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 2.460SEC
FUER DAS LOESEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 0.660SEC
Sollen die Potentiale ausgegeben werden? Ja/Ne NE
Soll das Netz und das Geschwindigkeitsfeld geplottet werden?
Ja/Ne JA

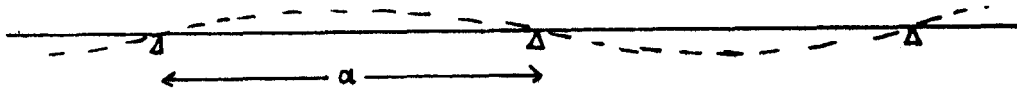
```

Nummer der Plotdatei? 2  
 Fuer das Erstellen der Plotdatei  
 verbrauchte CPU-Zeit: 4.940SEC  
 CPU-Zeit gesamt: 8.760 SEC  
 FORTRAN STOP

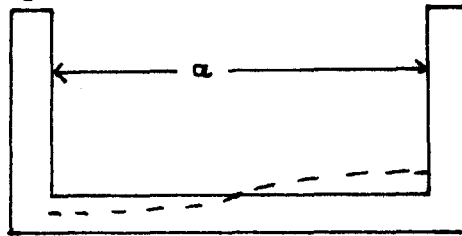
Das entstandene Bild 2 ist ebenfalls im Anhang 1.

#### 4.1.2 Generierung und Kontrolle des Netzes für die Strömung an einer schwingenden Platte

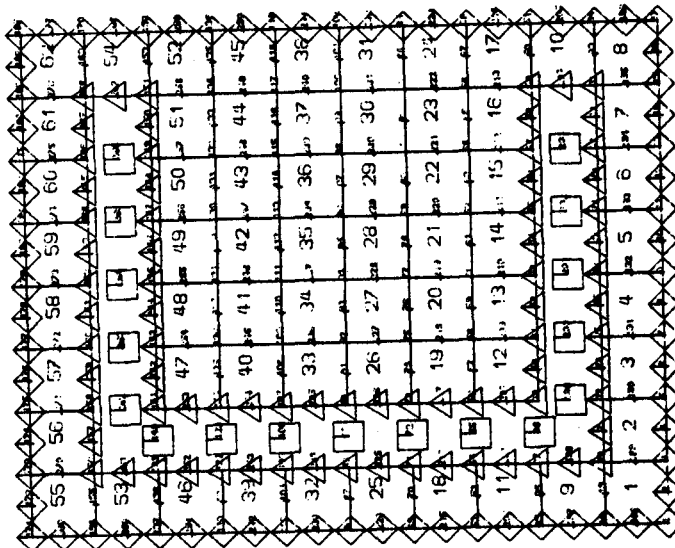
Das Problem der Berechnung der Strömung an einer schwingenden Platte soll hier gezeigt werden, weil es die Begrenzungen des vorgestellten Programms zeigt. Es wird eine unendlich ausgedehnte Platte, die im Abstand  $a$  unterstützt und nur von einer



Seite benetzt ist, betrachtet. Da mit dem beschriebenen Verfahren grundsätzlich nur endlich grosse und vollständig in die Flüssigkeit getauchte Körper berechnet werden können muß man hier ein etwas abgewandeltes Modell der Platte untersuchen. Man erwartet, daß das Potential periodisch mit dem Knotenabstand  $a$  ist und daß die Strömungsgeschwindigkeit über den Schwingungsbäuchen keine X-Komponente hat. Damit wird die Strömung an der Plattenoberfläche wohl gut durch die Umströmung des folgenden Körpers angenähert.



Die Generierung des Netzes mit dem Unterprogramm RASTGEN soll hier gezeigt werden: Mit Hilfe eines einfachen Programms lassen sich die Daten eines regelmäßigen Rechteckrasters leicht erzeugen. Statt der Dateiliste hier ein mit RASTGEN erzeugtes Plottbild des Netzes:



Gibt man jetzt noch zusätzlich an welche Knoten auf der Körperoberfläche liegen und welche Knoten nicht benutzt werden so kann man in einem Lauf mit INTAKTPOT die Daten eines berechenbaren Netzes erzeugen:

#### Programm zur Berechnung von Potentialströmungen

Sollen die Netzdaten aus digitalisierten Daten Der Knoten an der Koerperoberflaeche erzeugt werden? Ja/Ne NE

Name der Eingabedatei? PLATTE.DAT

Soll das Netz aus einem Raster generiert werden? Ja/Ne JA

Soll das Raster geplottet werden? Ja/Ne JA

Nummer der Plotdatei? 1 (Plot der Eingabedatei)

Im folgenden brauchen nur Zahlen eingegeben zu werden wo Abweichungen von den angegebenen gewünscht werden, sonst <RET>

X-Koordinate der Rotationsachse= 0.000? <RET>

Y-Koordinate der Rotationsachse= 0.000? <RET>

Winkelgeschwindigkeit (Bogenmass)= 0.000? <RET>

X-Komponente der Anstroemgeschwindigkeit= 0.000? <RET>

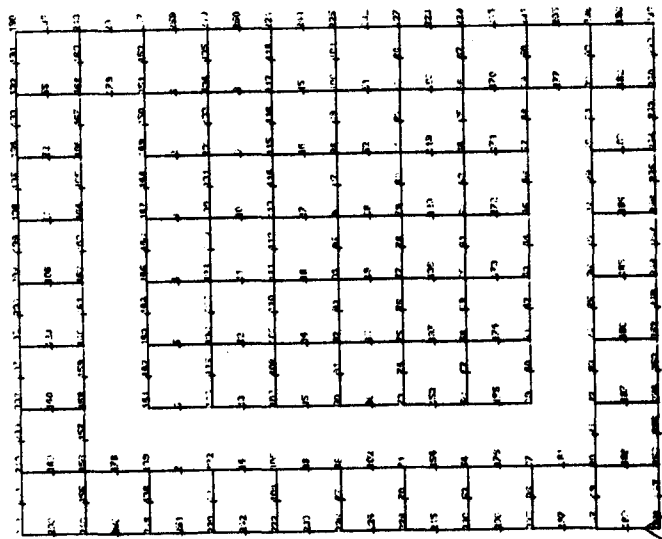
Y-Komponente der Anstroemgeschwindigkeit= 0.000? <RET>

Sollen die Netzdaten gespeichert werden? Ja/Ne JA

Dateiname? PLATTNETZ.DAT

Fuer das Lesen und Generieren  
verbrauchte CPU-Zeit: 20.310SEC

Der Plot der Eingabedatei ist im nächsten Bild wiedergegeben.



Die weitere Verarbeitung des Netzes ist in 4.2.2 gezeigt.

## 4.2 Anwendung von HYDMAS

### 4.2.1 Hydrodynamische Masse der Ellipse

Die Daten der in 4.1 beschriebenen Ellipse waren mit SCHREIBEN auf die Datei NETZ.DAT geschrieben worden. Diese Daten können verwendet werden um die hydrodynamische Massenmatrix für die Starrkörperbewegung in den drei Freiheitsgraden (X-Translation, Y-Translation und Rotation um den Mittelpunkt) zu Berechnen. Die Ansatzfunktionen für die Starrkörperbewegungen Q1, Q2 und Q3 sind im Anhang angegeben.

Das Ergebnis einer Berechnung mit HYDMAS ist dann folgendes:

```

FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 2.510SEC
FUER DAS LOESEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 0.700SEC
FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 2.300SEC
FUER DAS LOESEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 0.700SEC
FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 2.200SEC
FUER DAS LOESEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 0.640SEC
  63.57566071      -0.00001583      -0.00001681
 -0.00001857      254.36470032      0.00015536
 -0.00003147      0.00033772      1446.82006836

```

Die unten gedruckte Matrix ist die Matrix der hydrodynamischen Massen. Nach /6/ ist das analytische Ergebnis folgendes:

```

  64.      0.      0.
  0.      254.      0.
  0.      0.      1449.

```

#### 4.2.2 Hydrodynamische Masse einer schwingenden Platte

Das Problem und der Lösungsansatz ist bereits in 4.1.2 dargestellt. Das benötigte Berechnungsnetz befindet sich auf der Datei PLATTNETZ.DAT. Zur Formulierung des Verschiebungsansatzes:

Die Platte ist die Körperkante zwischen dem Knoten mit den Koordinaten (4,4) und (4,16). Da der Verschiebungsansatz  $(V_x, V_y)$  für den ganzen Körper definiert werden muß ergibt sich folgendes:

$(V_x, V_y) = (0, 0)$  , für  $X=4$  oder  $y < 4$  oder  $y > 16$   
 $(V_x, V_y) = (\cos((X-4)/12), 0)$  , für  $X=4$  und  $4 < Y < 16$

In FORTRAN sieht dieser Verschiebungsansatz, Unterprogramm Q1 dann so aus:

```

      SUBROUTINE Q1(RES,X)
C VERSCHIEBUNGSANSATZ FUER PLATTENSCHWINGUNG
      REAL RES(2),X(2)
      PI = 3.1415927
      RES(1) = 0.0
      RES(2) = 0.0
      IF ((X(1).LT.3.9).OR.(X(1).GT.4.1)) RETURN
      IF ((X(2).LT.3.9).OR.(X(1).GT.16.1)) RETURN
      RES(1) = COS(PI*(X(2)-4.0)/12.)
      RETURN
      END

```

In HYDMAS und PINT muß der Parameter IANS=1 gesetzt werden, weil nur ein Verschiebungsansatz benutzt wird. Zusätzlich wurde in HYDMAS nach dem Aufruf von POT2D noch ein Aufruf von MALEN eingefügt um ein Plottbild des Netzes und der Strömung zu erhalten. HYDMAS liefert folgendes Ergebnis:

```

FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSSYSTEMS
VERBRAUCHTE CPU-ZEIT: 10.480SEC
FUER DAS LOESEN DES GLEICHUNGSSYSTEMS

```

VERBRAUCHTE CPU-ZEIT: 49.380SEC

22.87329102

Nach /6/ ist die hydrodynamische Masse einer einseitig benetzten schwingenden Platte bezogen auf die Breite der Platte und auf die Dichte der Flüssigkeit für den Knotenabstand 12 analytisch 22.2. Es wird also auch hier eine recht gute Übereinstimmung erzielt. Der Plot ist Bild 3 im Anhang 1. Man sieht, daß die Strömungsgeschwindigkeit bis zur Oberkante der seitlichen Begrenzungen weitgehend abgeklungen ist.

#### 4.3 Berechnung der Strömung um einen Kreiszylinder mit verschiedenen Netzen

In den vorher gezeigten Beispielen wurden stets isoparametrische Elemente mit 8 Knoten benutzt, weil dies bezüglich der Rechenzeit und Genauigkeit günstig scheint. Zur Demonstration soll hier das Beispiel der Umströmung eines Kreiszylinders benutzt werden:

Auf der Oberfläche befinden sich 16 Knoten. Es wurden verschiedene Elementtypen benutzt und der maximale Abweichung eines Knotenpotentials von der analytischen Lösung /7/ für einen Kreiszylinder mit gleichem Radius bezogen auf das maximal zu erwartende Potential ermittelt. Das Ergebnis ist in der folgenden Tabelle wiedergegeben:

Zwischenknoten	Anzahl der Elemente	Anzahl der Knoten pro Element	CPU-Zeit [sec]	Maximaler Fehler [%]
innen, seitlich, außen	8	8	1.2	.32
keine	16	4	1.9	4.9
außen	16	5	2.2	1.9
seitlich	16	6	2.0	3.7
innen	8	5	1.0	10.
außen, innen	8	6	1.2	4.1
seitlich, außen	16	6	2.5	.67

Diese Ergebnisse entsprechen bezüglich der Rechenzeit den Erwartungen, denn bei einem so kleinen System ist die für die Lösung des Gleichungssystems benötigte Zeit viel geringer als die zur Berechnung der Koeffizienten des Gleichungssystems. Diese Zeit nun ist im wesentlichen abhängig von der Anzahl der Elemente, der Anzahl der Elemente auf der Körperoberfläche und der Anzahl der Knoten auf dem Außenrand. Es war daher zu erwarten, daß Netze mit Zwischenknoten an der Körperoberfläche besonders geringe Rechenzeit benötigen würden, da hier die Anzahl der Elemente nur 1/2 mal so groß ist wie bei den einfacheren Elementen. Prinzipiell kann man auch erkennen, daß die Genauigkeit mit zunehmender Anzahl von Knoten pro Element steigt, weil in jeder Richtung parabolische Funktionen den Verlauf des Potentials wohl am besten annähern. Außerdem tritt bei Elementen, die auf der Körperoberfläche keine Zwischenknoten haben ein großer systematischer Fehler dadurch auf, daß das Berechnungsmodell des Kreiszylinders ein einbeschriebenes 16-Eck ist, so daß der zur Berechnung benutzte Körper kleiner ist als der Kreiszylinder.

Literatur

- /1/ Lenoir, Jami  
A Variational Formulation for Exterior Problems in  
Linear Hydrodynamics  
Computer Methods in Applied Mechanics and Engineering  
North Holland Publishing Company 1978
  
- /2/ Jensen  
Eine Finite-Elemente-Methode zur Berechnung von  
Potentialströmungen  
Diplomarbeit, Institut für Schiffbau 1983
  
- /3/ Bronnstein, Semendjajew  
Taschenbuch der Mathematik  
Verlag Harri Deutsch, Zürich u. Frankfurt/Main 1975
  
- /4/ Fletcher  
Computational Galerkin Methods  
Springer Verlag, New York 1984
  
- /5/ Bathe, Wilson  
Numerical Methods in Finit Element Analysis  
Prentice Hall Inc., Englewood Cliffs, New Jersey 1976
  
- /6/ Söding  
Vibrationen von Schiffen I  
Vorlesungsmanuskript Nr. 23  
Institut für Schiffbau, 1983
  
- /7/ Wieghardt  
Theoretische Strömungslehre  
B.G.Teubner, Stuttgart 1974

Anhang 1

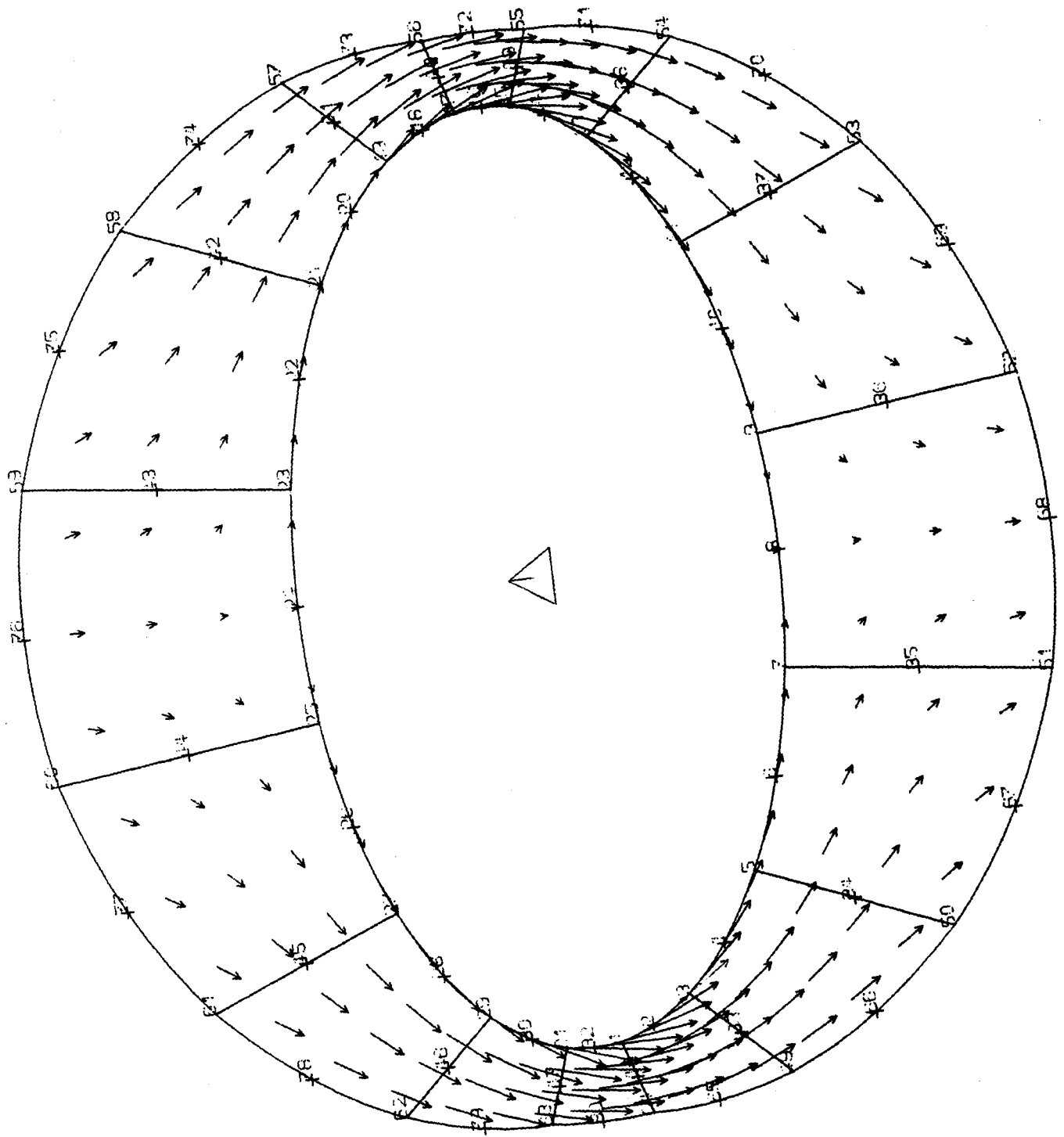


Bild 1

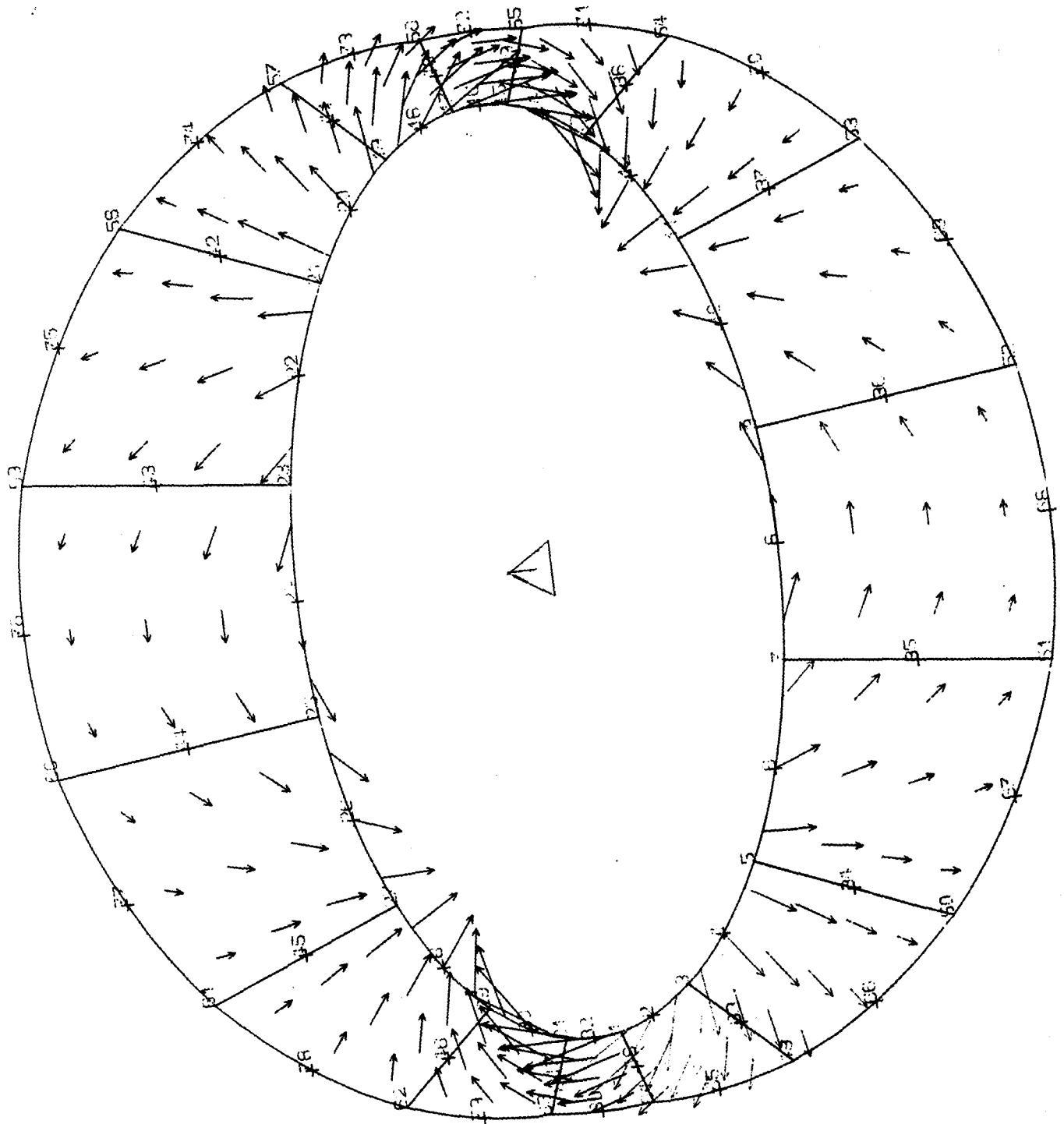


Bild 2

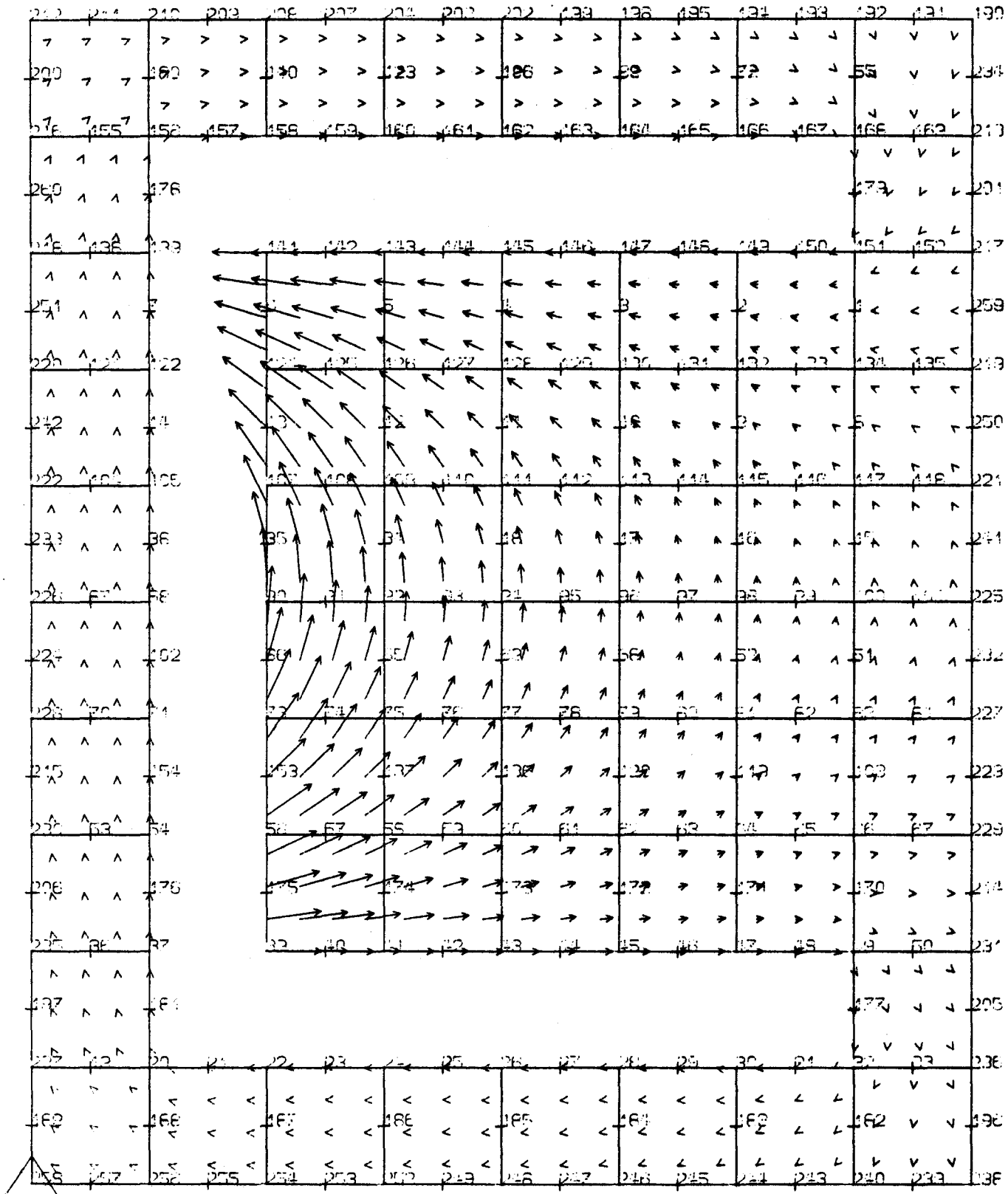


Bild 3

Anhang 2. Programmlisten

```

SUBROUTINE POT2D(PK,KN,NK,NE,ZEIGER,BODKN,SIGKN,NKS,GV,
1 RI,IAD,KPR)
C UNTERPROGRAMM ZUR BERECHNUNG DER POTENTIALSTROEMUNG UM
C BELIEBIGE KOERPER (2-DIMENSIONAL) MITTELS
C FINITE ELEMENTE ANSATZ NACH JAMI/LENOIR 1979
C ES WERDEN 4..8 KNOTIGE ISOPARAMETRISCHE ELEMENTE
C NACH BATHE/WILSON BENUTZT
C
C LETZTE AENDERUNG 29.2.84
C
C BEDEUTUNG DER PARAMETER:
C RI          REAL-FELD, MINDESTGROESSE IFELD (SIEHE UNTEN)
C            MUSS BEI AUFRUF VOLLSTAENDIG MIT NULLEN BESETZT SEIN
C            ENTHAELT NACH PROGRAMMAUSFUERUNG DIE KNOTENPOTENTIALE
C IAD         NUMMER DER AUSGABEDATEI
C EPR         TRUE:      DIE EINGABEDATEN WERDEN AUF IAD GESCHRIEBEN
C KPR         TRUE:      DIE KOEFFIZIENTEN DES GLEICHUNGSSYSTEMA WERDEN
C            AUF IAD GESCHRIEBEN
C PK(I,J)    KOORDINATEN DES KNOTEN J
C            MIT I=1 X-KOORDINATE
C            UND I=2 Y-KOORDINATE
C KN(I,J)    KNOTENNUMMER DES I-TEN KNOTEN DES
C            J-TEN ELEMENTS
C ZEIGER(I)  ZEIGER DES I-TEN KNOTEN
C BODKN(J)   NUMMER DES J-TEN KNOTEN AUF DER
C            KOERPEROBERFLAECHE
C SIGKN(K)   NUMMER DES K-TEN KNOTEN AUF DEM
C            RAND DES BERECHNUNGSRAUMS
C            ZUSAMMENHANG ZWISCHEN ZEIGER, BODKN UND SIGKN:
C            FUER KNOTEN AUF DER KOERPEROBERFLAECHE IST
C            ZEIGER>0 UND I=BODKN(ZEIGER(I))
C            FUER KNOTEN DIE SICH AN KEINEM DER RAENDER BEFINDEN
C            IST ZEIGER=0
C            FUER KNOTEN AUF DEM AUSSENRAND IST
C            ZEIGER<0 UND I=SIGKN(-ZEIGER(I))
C GV         IST DER VEKTOR DER GESCHWINDIGKEIT DER KOERPEROBERFLAECHE
C            GV(1,I)=X-KOMPONENTE AM KNOTEN BODKN(I)
C            GV(2,I)=Y-KOMPONENTE AM KNOTEN BODKN(I)
C RKIJ(I,J)  IST DER KOEFFIZIENT DES GLEICHUNGSSYSTEMS
C            IN ZEILE J, SPALTE I
C RI(I)      IST DIE I-TE RECHTE SEITE
C NKS        ANZAHL DER KNOTEN AUF DEM AUSSENRAND SIGMA
C
C PARAMETER ZUR BESTIMMUNG DER FELDGRUESSEN
C SOLL>= ANZAHL ALLER KNOTEN SEIN
C            PARAMETER (IFELD=300)
C
C LOGICAL KPR,RPR
C REAL RKIJ(IFELD,IFELD),GG(IFELD/2),PK(2,1),RI(1),GV(2,1)
C REAL IWDG(IFELD/2,IFELD/2),IWG(3,IFELD/2)
C INTEGER ZEIGER(1),BODKN(IFELD/2),SIGKN(1),KN(8,1)
C CALL CPUTIM(CP,CT)
C INITIALISIERUNG DER FELDER RKIJ,GG,IWDG
C CALL NULLMACH(GG,IFELD/2)

```

```

      CALL NULLMACH(IWDG, (IFELD/2)**2)
      CALL NULLMACH(RKIJ, IFELD*NK)
      CALL GLSY(RKIJ, RI, NK, PK, NE, KN, ZEIGER, BODKN, SIGKN, GV, GG, IWDG, IFELD,
1IWG)
      CALL CPUTIM(CP, CT)
      CALL TEXPRI('FUER DIE BERECHNUNG DER KOEFFIZIENTEN DES GLEICHUNGSS
1SYSTEMS', 6)
      WRITE(6, 1) CP
      IF (KPR) THEN
          CALL TEXPRI('KOEFFIZIENTENMATRIX', IAD)
          CALL MAPRINT(RKIJ, NK-NKS, NK-NKS, IFELD, IAD)
          CALL TEXPRI('RECHTE SEITE', IAD)
          CALL MAPRINT(RI, NK-NKS, 1, IFELD, IAD)
          CALL CPUTIM(CP, CT)
          CALL TEXPRI('FUER DAS DRUCKEN DES GLEICHUNGSSYSTEMS', 6)
          WRITE(6, 1) CP
      END IF
      CALL SOLVE(NK-NKS, RKIJ, RI, IFELD)
      CALL CPUTIM(CP, CT)
      CALL TEXPRI('FUER DAS LOESEN DES GLEICHUNGSSYSTEMS', 6)
      WRITE(6, 1) CP
      CALL RANDPOT(RI, IWDG, GG, NK, ZEIGER, IFELD)
      RETURN
1  FORMAT(' VERBRAUCHTE CPU-ZEIT:', F7.3, 'SEC')
      END

      SUBROUTINE GLSY(RKIJ, RI, NK, PK, NE, KN, ZEIGER, BODKN, SIGKN, GV,
1GG, IWDG, IFELD, IWG)
      C UNTERPROGRAMM ZUR ERSTELLUNG DER STEIFIGKEITSMATRIX UND DER RECHTEN SEITEN
      C LETZTE AENDERUNG 5.3.84
      REAL X(2, 8), IGG(8, 8), PK(2, 1), RKIJ(IFELD, IFELD)
      REAL GG(1), RI(1), GV(2, 1), IWDG(IFELD/2, IFELD/2), IWG(1)
      INTEGER KN(8, 1), ZEIGER(1), BODKN(1), SIGKN(1), KK(8)
      C BERECHNUNG ALLER INTEGRALE UEBER DIE KOERPEROBERFLAECHE
      CALL OINT(IWDG, GG, RI, ZEIGER, SIGKN, BODKN, KN, PK, NK, NE, GV, IFELD,
1IWG)
      C AUFADDIEREN ALLER INTEGRALE UEBER DEN BERECHNUGSRAUM
      DO 100 J=1, NE
          CALL ELEDAT(J, KN, PK, X, KK)
      C BERECHNUNG VON INTEGRAL(GRADIEND(Wi)*GRADIENT(Wj))
      CALL INTGRAD(X, KK, IGG)
      C INTEGRAL(GRADIENT(EXTENSION(INTEGRAL(G*GREEN) dS))*GRADIENT W)d OMEGA
      DO 30 K=1, 8
          IF (KK(K).EQ.D) GOTO 30
          IF (ZEIGER(KK(K)).LT.D)
1          CALL GGADD(RI, KK, K, IGG, -ZEIGER(KK(K)), GG)
30          CONTINUE
      DO 50 I=1, NK
          IF (BODKN(I).EQ.D) GOTO 60
      DO 50 K=1, 8
          IF (KK(K).EQ.D) GOTO 50
          IF (ZEIGER(KK(K)).LT.D) THEN
      DO 40 L=1, 8
          IF (KK(L).EQ.D) GOTO 40
          RKIJ(KK(L), BODKN(I))=RKIJ(KK(L), BODKN(I))+
1 IGG(K, L)*IWDG(-ZEIGER(KK(K)), I)

```

```

40     CONTINUE
      END IF
50     CONTINUE
60     CONTINUE
C FUER ALLE ELEMENTE
  DO 100 I=1,8
    IF(KK(I))100,100,80
80    DO 100 K=1,8
      IF(KK(K))100,100,90
90    RKIJ(KN(I,J),KN(K,J))=RKIJ(KN(I,J),KN(K,J))+IGG(I,K)
100   CONTINUE
      RETURN
      END

```

```

      SUBROUTINE MAPRINT(M,NZ,NS,IM,IDAT)
C UNTERPROGRAMM ZUM SCHREIBEN EINER MATRIX M(ZEILE,SPALTE) MIT NZ ZEILEN
C UND NS SPALTEN AUF DIE DATEI IDAT
C IM= MAXIMALGROESSE VON M IN DER DIMENSIONANWEISUNG
C LETZTE AENDERUNG 2.1.84
      REAL M(1)
      DO 100 I=1,NZ
100   WRITE(IDAT,1)(M((J-1)*IM+I),J=1,NS)
      RETURN
      1 FORMAT(BF16.8)
      END

```

```

      SUBROUTINE SOLVE(NK,RKIJ,RI,IFELD)
C UNTERPROGRAMM ZUM LOESEN DES GLEICHUNGSSYSTEMS MIT HILFE DES UNTERPROGRAMMS
C SIMQ2 SOWIE ZUM AUSDRUCKEN DER LOESUNG, FALLS IDRUCK>0
C IFELD IST DIE DIMENSION DER KOEFFIZIENTENMATRIX
C LETZTE AENDERUNG 29.2.84
      REAL RKIJ(IFELD,IFELD),RI(1)
      DO 10 I=1,NK
10    RKIJ(I,NK+1)=-RI(I)
      CALL SIMQ2(RKIJ,NK,1,IFELD,KS,.0001)
      IF (KS.EQ.4) THEN
        CALL GROSAT('DAS GLEICHUNGSSYSTEM IST SINGULAER',6)
        STOP
      END IF
      DO 20 I=1,NK
20    RI(I)=RKIJ(I,NK+1)
      RETURN
      END

```

```

      SUBROUTINE RANDPOT(POT,IWDG,GG,NK,ZEIGER,IFELD)
C UNTERPROGRAMM ZUR BERECHNUNG DES POTENTIALS AN DEN AUSSENRANDKNOTEN
C (ZEIGER<0) AUS DEM POTENTIAL AN DEN KOERPERKNOTEN (ZEIGER>0)
C ES WERDEN DIE MATRIX IWDG UND DER VEKTOR GG BENOETIGT. SIEHE OINT
C LETZTE AENDERUNG 30.1.84
      REAL POT(1),GG(1),IWDG(IFELD/2,IFELD/2)
      INTEGER ZEIGER(1)
      DO 100 I=1,NK
      IF (ZEIGER(I).GE.0) GOTO 100
      POT(I)=-GG(-ZEIGER(I))
100

```

```

DO 100 J=1,NK
IF (ZEIGER(J).LE.0) GOTO 100
POT(I)=POT(I)+POT(J)*IWDG(-ZEIGER(I),ZEIGER(J))
100 CONTINUE
RETURN
END

```

```

SUBROUTINE OINT(IWDG,GG,RI,ZEIGER,SIGKN,BODKN,KN,PK,NK,NE,GV,
1IFELD,IWG)
C UNTERPROGRAMM ZUR BERECHNUNG VON INTEGRALEN UEBER DIE KOERPEROBERFLAECHE
C MATRIX IWDG( RANDKNOTEN, KOERPERKNOTEN) ENTHAELT
C INTEGRAL(ABLEITUNG DER GREENFUNKTION(RANDKNOTEN,P) NACH DER
C KOERPERNORMALEN(P)*ANSATZFUNKTION FUER EINEN KOERPERKNOTEN,
C BERECHNET AN DER STELLE P)
C VEKTOR GG( RANDKNOTEN) ENTHAELT INTEGRAL(GREENFUNKTION(RANDKNOTEN,P)
C *NORMALGESCHWINDIGKEIT DES KOEPERS BEI P)
C VEKTOR RI(KNOTEN) ENTHAELT INTEGRAL( NORMALGESCHWINDIGKEIT DES
C KOEPERS * ANSATZFUNKTION FUER DEN KNOTEN)
C
C LETZTE AENDERUNG 6.3.84
INTEGER ZEIGER(1),SIGKN(1),BODKN(1),KN(8,1),KK(8)
REAL IWG(3,1),IWDG(IFELD/2,IFELD/2),X(2,8),PK(2,1)
REAL RI(1),GG(1),DRI(3),GV(2,1),GVE(2,8)
DO 100 I=1,NE
IF (ZEIGER(KN(3,I)).LE.0) GOTO 100
IF (ZEIGER(KN(4,I)).LE.0) GOTO 100
CALL ELEDAT(I,KN,PK,X,KK)
DO 10 J=1,8
IF(KK(J).EQ.0) GOTO 10
IF (ZEIGER(KK(J)).GT.0) THEN
GVE(1,J)=GV(1,ZEIGER(KK(J)))
GVE(2,J)=GV(2,ZEIGER(KK(J)))
ELSE
GVE(1,J)=0.
GVE(2,J)=0.
END IF
10 CONTINUE
CALL NULLMACH(IWG,3*(IFELD/2))
CALL ELOINT(X,KN,PK,SIGKN,NK,IWG,GG,DRI,GVE,1)
CALL OINADD(KK(3),KK(4),KK(7),ZEIGER,SIGKN,DRI,
1 RI,IWG,IWDG,IFELD,NK)
IF (ZEIGER(KK(1)).LE.0) GOTO 100
CALL NULLMACH(IWG,3*(IFELD/2))
CALL ELOINT(X,KN,PK,SIGKN,NK,IWG,GG,DRI,GVE,2)
CALL OINADD(KK(4),KK(1),KK(8),ZEIGER,SIGKN,DRI,
1 RI,IWG,IWDG,IFELD,NK)
100 CONTINUE
END

```

```

SUBROUTINE ELOINT(X,KN,PK,SIGKN,NK,IWG,GG,DRI,GVE,IRL)
C UNTERPROGRAMM ZUR BERECHNUNG VON INTEGRAL UEBER DEN RAND DES
C ELEMENTS AN DEM R(IRC)=1 IST.
C WENN IRL=1 WIRD ENTLANG DES RANDES MIT DEN
C KNOTEN 3-(7)-4 INTEGRIERT
C WENN IRL=2 WIRD ENTLANG DES RANDES MIT DEN

```

```

C      KNOTEN 4-(8)-1 INTEGRIERT
C
C MATRIX IWG:  Hi * ABLEITUNG VON GREENSFUNKTION(M,P) NACH DER
C      NORMALEN AUF DIE STRECKE ̄HIER IST P̄;DIE PUNKTE M SIND
C      DIE AUSSENRANDKNOTEN.
C VEKTOR GG( RANDKNOTEN):  GREENFUNKTION(RANDKNOTEN M, P)
C      *NORMALGESCHWINDIGKEIT DES KOERPERS BEI P)
C VEKTOR DRI:  NORMALEGESCHWINDIGKEIT * ANSATZFUNKTION
C DIE KOORDINATEN DER PUNKTE M SIND IM FELD PK GESPEICHERT.
C DIE NORMALE WEISST NACH RECHTS, WENN MAN VON 4 NACH 1 BLICKT.
C LETZTE AENDERUNG 6.3.84
      REAL X(2,8),XM(2),IWG(3,1),R(2),DH(2),JM(2,2),XP(2),PK(2,1)
      REAL H(8),DHD(2,8),DRI(3),GG(1),DG(2),GVE(2,8),GVXP(2)
      INTEGER SIGKN(1),KK(8),K(3,2)
      COMMON/GAUSS/STELLE(5),GEW(5),NSTUETZ
C NUMMERN DER KNOTEN,DIE AN DER ZU DURCHLAUFENDEN KANTE LIEGEN
      DATA K/3,4,7,4,1,8/
C INITIALISIERUNG
      DO 10 I=1,3
10  DRI(I)=0
      R(1)=1.
      R(2)=-1.
C INTEGRATION NACH GAUSS
      DO 100 I=1,NSTUETZ
      R( IRL)=STELLE(I)*2-1
      FAK=GEW(I)*2
      CALL ANSATZ(R,KK,H,DHD)
      CALL JACOBI(X,DHD,JM)
      CALL DBOG(DS,JM,IRL)
      CALL XINTER(GVXP,GVE,H)
      CALL NORMKO(G,GVXP,JM,IRL)
      DRI(1)=DRI(1)+G*H(K(1,IRL))*FAK*DS
      DRI(2)=DRI(2)+G*H(K(2,IRL))*FAK*DS
      DRI(3)=DRI(3)+G*H(K(3,IRL))*FAK*DS
      CALL XINTER(XP,X,H)
      DO 50 J=1,NK
      IF (SIGKN(J).EQ.0) GOTO 100
      CALL GREEN(GRE,DG,PK(1,SIGKN(J)),XP)
      CALL NORMKO(DGDN,DG,JM,IRL)
      GG(J)=GG(J)+GRE*G*FAK*DS
      IWG(1,J)=IWG(1,J)+H(K(1,IRL))*DGDN*DS*FAK
      IWG(2,J)=IWG(2,J)+H(K(2,IRL))*DGDN*DS*FAK
50  IF (KK(K(3,IRL)).GT.0) IWG(3,J)=IWG(3,J)+H(K(3,IRL))*DGDN*DS*FAK
100 CONTINUE
      RETURN
      END

```

```

      SUBROUTINE INTGRAD(X,KK,IGG)
C UNTERPROGRAMM ZURBERECHNUNG VON INTEGRAL UEBER DIE GESAMTE
C ELEMENTFLAECHE VON GRADIENT(Hi)*GRADIENT(Hj) FUER ALLE i,j
C ANSATZFUNKTIONEN
C LETZTE AENDERUNG: 27.2.84
      REAL X(2,8),IGG(8,8),JM(2,2),R(2),H(8),DHD(2,8)
      INTEGER KK(8)
      COMMON/GAUSS/STELLE(5),GEW(5),NSTUETZ
C INITIALISIERUNG

```

```

      DO 10 I=1,8
      DO 10 J=1,8
      10 IGG(I,J)=0
C INTEGRATION NACH GAUSS
      DO 100 I=1,NSTUETZ
      R(1)=STELLE(I)*2-1
      DO 100 J=1,NSTUETZ
      R(2)=STELLE(J)*2-1
      FAK=GEW(I)*GEW(J)*4
C BERECHNUNG DER ANSATZFUNKTIONEN
      CALL ANSATZ(R, KK, H, DHD)
C BERECHNUNG DER JACOBISCHEN MATRIX
      CALL JACOBI(X, DHD, JM)
      CALL OMEGA(DOM, JM)
      CALL KOTRANS(DHD, JM)
      DO 50 K=1,8
      DO 50 L=1,8
      50 IGG(K,L)=IGG(K,L)+SKALP(DHD(1,K), DHD(1,L), 2)*FAK*DOM
100 CONTINUE
      RETURN
      END

```

```

      SUBROUTINE ANSATZ(R, KK, H, DHD)
C BERECHNUNG DER ANSATZFUNKTIONEN UND IHRER ABLEITUNGEN FUER DAS
C ZWEIDIMENSIONALE ISOPARAMETRISCHE ELEMENT NACH BATHE/WILSON
C LETZTE AENDERUNG 12.1.84
C
C EINGANGSVARIABLEN:
C R(1)  LOKALE KOORDINATE R
C R(2)  LOKALE KOORDINATE S
C KK    VEKTOR MIT 8 KOMONENTEN
C       WENN KK(I)>0 IST DER I-TE KNOTEN VORHANDEN
C       WENN KK(I)=0 IST DER I-TE KNOTEN NICHT VORHANDEN
C
C ERGEBNISSE
C H(I)  WERT DER I-TEN ANSATZFUNKTION BEI (R(1),R(2))
C DHD(J,I) WERT DER ABLEITUNG NACH R(J) DER I-TEN ANSATZFUNKTION
C        BEI (R(1),R(2))
C
      INTEGER KK(8)
      REAL R(2), H(8), DHD(2,8)
      DO 100 I=5,8
      H(I)=0.
      DHD(1,I)=0.
100 DHD(2,I)=0.
      IF (KK(5))5,5,1
      1 H(5)=.5*(1-R(1)**2)*(1+R(2))
      DHD(1,5)=-R(1)*(1+R(2))
      DHD(2,5)=.5*(1-R(1)**2)
      5 IF (KK(6))6,6,2
      2 H(6)=.5*(1-R(2)**2)*(1-R(1))
      DHD(1,6)=-.5*(1-R(2)**2)
      DHD(2,6)=-R(2)*(1-R(1))
      6 IF (KK(7))7,7,3
      3 H(7)=.5*(1-R(1)**2)*(1-R(2))
      DHD(1,7)=-R(1)*(1-R(2))

```

```

DHD(2,7)=-.5*(1-R(1)**2)
7 IF(KK(8))8,8,4
4 H(8)=-.5*(1-R(2)**2)*(1+R(1))
DHD(1,8)=.5*(1-R(2)**2)
DHD(2,8)=-R(2)*(1+R(1))
8 H(1)=.25*(1+R(1))*(1+R(2))-.5*(H(5)+H(8))
DHD(1,1)=.25*(1+R(2))-.5*(DHD(1,5)+DHD(1,8))
DHD(2,1)=.25*(1+R(1))-.5*(DHD(2,5)+DHD(2,8))
H(2)=.25*(1-R(1))*(1+R(2))-.5*(H(5)+H(6))
DHD(1,2)=-.25*(1+R(2))-.5*(DHD(1,5)+DHD(1,6))
DHD(2,2)=.25*(1-R(1))-.5*(DHD(2,5)+DHD(2,6))
H(3)=.25*(1-R(1))*(1-R(2))-.5*(H(6)+H(7))
DHD(1,3)=-.25*(1-R(2))-.5*(DHD(1,6)+DHD(1,7))
DHD(2,3)=-.25*(1-R(1))-.5*(DHD(2,6)+DHD(2,7))
H(4)=.25*(1+R(1))*(1-R(2))-.5*(H(7)+H(8))
DHD(1,4)=.25*(1-R(2))-.5*(DHD(1,7)+DHD(1,8))
DHD(2,4)=-.25*(1+R(1))-.5*(DHD(2,7)+DHD(2,8))
RETURN
END

```

```

SUBROUTINE XINTER(XI,X,H)
C UNTERPROGRAMM ZUR INTERPOLATION DER KARTESISCHENKOORDINATEN
C LETZTE AENDERUNG 27.2.83
REAL XI(2),X(2,8),H(8)
DO 5 I=1,2
5 XI(I)=0.
DO 10 I=1,8
DO 10 J=1,2
10 XI(J)=XI(J)+H(I)*X(J,I)
RETURN
END

```

```

SUBROUTINE DBOG(DS,JM,IV)
C UNTERPROGRAMM ZUR BERECHNUNG DER LAENGE EINES
C BOGENELEMENTS dIV, MIT IV=1 FUER r VARIABLE UND
C IV=2 FUER S VARIABLE
C LETZTE AENDERUNG 19.12.83
REAL JM(2,2)
DS=SQRT(JM(IV,1)**2+JM(IV,2)**2)
RETURN
END

```

```

SUBROUTINE GREEN(GRE,DG,XM,XP)
C GREENFUNKTION EINER QUELLE IN UNBESCHRAENKTER FLUESSIGKEIT(2-D) GRE
C UND DER ABLEITUNGEN DER GREENFUNKTION(XM,XP) DG
C LETZTE AENDERUNG 16.1.84
REAL XM(2),XP(2),DG(2)
DATA PI/3.14159/
RMP2=(XM(1)-XP(1))**2+(XM(2)-XP(2))**2
DG(1)=(XP(1)-XM(1))/2/PI/RMP2
DG(2)=(XP(2)-XM(2))/2/PI/RMP2
GRE=ALOG(RMP2)/4/PI
RETURN
END

```

```

SUBROUTINE JACOBI(X,DHD,JM)
C UNTERPROGRAMM ZUR BERECHNUNG DER JACOBISCHEN MATRIX
C LETZTE AENDERUNG 27.2.84
  REAL X(2,8),DHD(2,8),JM(2,2)
  DO 50 I=1,2
  DO 50 J=1,2
50 JM(I,J)=0.
  DO 100 I=1,8
  DO 100 J=1,2
  DO 100 K=1,2
100 JM(J,K)=JM(J,K)+DHD(J,I)*X(K,I)
  RETURN
END

```

```

SUBROUTINE KOTRANS(DHD,JM)
C UNTERPROGRAMM WANDELT ABLEITUNGEN NACH DEN LOKALENKOORDINATEN
C IN SOLCHE NACH DEN GLOBALEN UM
C LETZTE AENDERUNG 12.1.84
  REAL DHD(2,8),JM(2,2),NM(2,2),NDET
  CALL DETERM(NDET,JM,2)
  DO 30 J=1,8
  DO 10 I=1,2
  NM(I,1)=DHD(I,J)
10 NM(I,2)=JM(I,2)
  CALL DETERM(XDET,NM,2)
  DO 20 I=1,2
  NM(I,2)=DHD(I,J)
20 NM(I,1)=JM(I,1)
  CALL DETERM(YDET,NM,2)
  DHD(1,J)=XDET/NDET
30 DHD(2,J)=YDET/NDET
  RETURN
END

```

```

SUBROUTINE OMEGA(DOM,JM)
C BERECHNET FAKTOR DOM ZUR UMRECHNUNG DES FLAECHEELEMENTES dA
C IN dr ds NACH BRONNSTEIN SEITE 221
C LETZTE AENDERUNG 12.1.84
  REAL JM(2,2)
  E=JM(1,1)**2+JM(1,2)**2
  F=JM(1,1)*JM(2,1)+JM(1,2)*JM(2,2)
  G=JM(2,1)**2+JM(2,2)**2
  DOM=SQRT(E*G-F**2)
  RETURN
END

```

```

SUBROUTINE MALEN
  1 (KN,PK,NK,NE,ZEIGER,POT,GSTAT,XPOL,GLCM,IGPLOT,IPD)
C MALT DAS NETZ, WENN IGPLOT=0
C MALT DAS NETZ UND DIE GESCHWINDIGKEITSVEKTOREN AN
C DER KOERPEROBERFLAECHE,WENN IGPLOT=1
C DER VEKTOR POT MUSS DANN ALLE KNOTENPOTENTIALE ENTHALTEN

```

```

C GSTAT IST DER VEKTOR DER STATIONAEREN ANSTROEMGESCHWINDIGKEIT
C GLCM IST DIE LAENGE IN CM EIN MIT GESCHWINDIGKEITSVEKTOR DER LAENGE
C 1 AUF DEM PLOT ERSCHEINEN SOLL.
C XPOL SIND DIE KOORDINATEN EINES SPEZIELL ZU MARKIERENDEN PUNKTES
C PLOTDATEI IST IPD
C LETZTE AENDERUNG 27.2.84
      REAL PK(2,1),POT(1),GSTAT(2),XD(2),XPOL(2)
      INTEGER KN(8,1),ZEIGER(1)
C     PK(1,I) X KOORDINATE VON I
C     PK(2,I) Y KOORDINATE VON I
      CALL DINA4(XD(1),XD(2),PK,NK,SCAL)
      CALL PLOTS(0,0,IPD)
      CALL PLOT(XD(1),XD(2),-3)
      CALL SYMBOL(XPOL(1)*SCAL,XPOL(2)*SCAL,1.,2,0.,-1)
      CALL NETZPLO(KN,PK,NK,NE,SCAL)
      IF (IGPLOT.EQ.0) THEN
        CALL ELENUM(KN,PK,NK,NE,SCAL)
      ELSE
        CALL NEWPEN(2)
        CALL GESCHPLO(POT,KN,PK,NE,ZEIGER,GSTAT,GLCM,SCAL)
      END IF
      CALL PLOT(0,0,999)
      RETURN
      END

```

```

      SUBROUTINE DINA4(XD,YD,PK,NK,SCAL)
C     UNTERPROGRAMM ERMITTELT MASSTABFAKTOR SCAL
C     SO,DASS A4 BLATT GEFUELLT WIRD, WENN PUNKTKOORDINATEN
C     MIT SCAL MULTIPLIZIERT WERDEN.
C     XD,YD KOORDINATEN DES ZU VERWENDENDEN URSPRUNGS
C     ERSTELLT VON MAX HEINEMANN
C     LETZTE AENDERUNG 22.2.84
      REAL PK(2,1)
C     FESTSTELLEN DER AUSDEHNUNG DES NETZES
      CALL RMAX(XMAX,XMIN,YMAX,YMIN,PK,NK)
      BMAX=XMAX-XMIN
      HMAX=YMAX-YMIN
      SCAL=AMIN1(24/HMAX,17/BMAX)
C     FESTLEGEN DES URSPRUNGS
      XD=-XMIN*SCAL+2.5
      YD=-YMIN*SCAL+3.
      RETURN
      END

```

```

      SUBROUTINE NETZPLO(KN,PK,NK,NE,SCAL)
C     MALT DAS NETZ
C     ERSTELLT VON MAX HEINEMANN
C     LETZTE AENDERUNG 26.1.84
      REAL PK(2,1)
      INTEGER KN(8,1)
C     PK(1,I) X KOORDINATE VON I
C     PK(2,I) Y KOORDINATE VON I
      DO 100 I=1,NE
100  CALL ELEPLOT(KN,PK,NK,I,SCAL)
      CALL NEWPEN(3)

```

```

DO 200 I=1,NK
200 CALL NUMBER(PK(1,I)*SCAL,PK(2,I)*SCAL,0.20,float(i),0.,-1)
RETURN
END

```

```

SUBROUTINE ELENUM(KN,PK,NK,NE,SCAL)
C NUMERIERT ELEMENTE
C ERSTELLT VON MAX HEINEMANN
C LETZTE AENDERUNG 30.12.83
REAL PK(2,1)
INTEGER KN(8,1)
CALL NEWPEN (2)
DO 100 I=1,NE
X=0
Y=0
DO 50 J=1,4
X=PK(1,KN(J,I))+X
Y=PK(2,KN(J,I))+Y
50 CONTINUE
X=X/4*SCAL
Y=Y/4*SCAL
CALL NUMBER (X,Y,0.4,FLOAT(I),0.,-1)
100 CONTINUE
RETURN
END

```

```

SUBROUTINE ELEPLOT(KN,PK,NK,IE,SCAL)
C UNTERPROGRAMM PLOTTET DAS ELEMENT IE
C LETZTE AENDERUNG 25.1.84
REAL XK(2,3),PK(2,1)
INTEGER KN(8,1)
DO 100 I=1,4
IF (I.LT.4) THEN
J=I+1
ELSE
J=1
END IF
K=I+4
DO 10 L=1,2
XK(L,1)=PK(L,KN(I,IE))*SCAL
10 XK(L,3)=PK(L,KN(J,IE))*SCAL
IF (KN(K,IE).GT.0) THEN
XK(1,2)=PK(1,KN(K,IE))*SCAL
XK(2,2)=PK(2,KN(K,IE))*SCAL
CALL KANTPLOT(XK)
ELSE
CALL LINE(XK(1,1),XK(2,1),2,4,0,0)
END IF
100 CONTINUE
RETURN
END

```

```

SUBROUTINE GESCHPLO(POT,KN,PK,NE,ZEIGER,GSTAT,GLCM,SCAL)
C UNTERPROGRAMM PLOTTET GESCHWINDIGKEITSPFEILE

```

```

C UND IN DER UNTEREN ELEMENTSCHICHT
C LETZTE AENDERUNG 28.2.84
  REAL PK(2,1),POT(1),X(2,8),H(8),DHD(2,8),U(2),XP(2),R(2)
  REAL ELPOT(8),JM(2,2),GSTAT(2)
  INTEGER KN(8,1),ZEIGER(1),KK(8)
  FAK=GLCM
  DO 100 J=1,NE
    CALL ELEDAT(J,KN,PK,X,KK)
    DO 10 I=1,8
      IF (KK(I).EQ.0) THEN
        ELPOT(I)=0.
      ELSE
        ELPOT(I)=POT(KK(I))
      END IF
    10 CONTINUE
  C BESTIMMUNG DER VERTEILUNG DER PFEILE IM ELEMENT
    R10=0.8
    R1S=0.6667
    R2S=0.6667
    R2U=-0.6667
  C FALLS KOERPERRAND SOLLEN INSBESONDERE AUCH HIER PFEILE GEPLOTTET WERDEN
    IF(ZEIGER(KK(3)).GT.0.AND.ZEIGER(KK(4)).GT.0) THEN
      R2U=-1.
      R2S=5./9.
    END IF
    IF(ZEIGER(KK(1)).GT.0.AND.ZEIGER(KK(4)).GT.0) THEN
      R10=1.
      R1S=5./9.
    END IF
  DO 100 R1=-0.66667,R10,R1S
    R(1)=R1
  DO 100 R2=R2U,.8,R2S
    R(2)=R2
  CALL ANSATZ(R,KN,H,DHD)
  CALL JACOBI(X,DHD,JM)
  CALL KOTRANS(DHD,JM)
  CALL UINTER(U,DHD,KN,ELPOT)
  CALL VEKADD(U,U,GSTAT,-1.,2)
  U(1)=U(1)*FAK
  U(2)=U(2)*FAK
  CALL XINTER(XP,X,H)
  XP(1)=XP(1)*SCAL
  XP(2)=XP(2)*SCAL
  CALL PFEIL(XP,U,.12)
100 CONTINUE
  RETURN
  END

```

SUBROUTINE UINTER(U,DHD,KN,ELPOT)

```

C UNTERPROGRAM ZUR BERECHNUNG DER STROEMUNGSGESCHWINDIGKEIT
C U AUS DEM POTENTIAL ELPOT AN DEN KNOTEN DES ELEMENTS
C DHD SIND DIE ABLEITUNGEN DER ANSATZFUNKTIONEN NACH X UND Y!!
C LETZTE AENDERUNG 26.1.84
  INTEGER KK(8)
  REAL DHD(2,8),U(2),ELPOT(8)
  DO 100 J=1,2

```

```

U(J)=0.0
DO 100 I=1,8
100 U(J)=U(J)+DHD(J,I)*ELPOT(I)
RETURN
END

```

PROGRAM HYDMAS

```

C PROGRAMM ZUR BERECHNUNG DER HYDRODYNAMISCHE MASSENMATRIX MITTELS
C DES UNTERPROGRAMMS POT2D.
C DER PARAMETER IFELD IN DER ROUTINE POT2D SOLLTE GENAU SO GROSS SEIN
C WIE IN DIESER ROUTINE.
C DATEI 5 MUSS HIERZU EINEN VOLLSTAENDIGEN DATENSATZ WIE FUER INTAKTPOT
C ENTHALTEN
C DER PARAMETER IANS MUSS IN DIESER ROUTINE SOWIE IN DEN ROUTINEN Q UND
C PINT AUF DIE DIE ZAHL DER VERWENDETEN ANSATZFUNKTIONEN GESETZT WERDEN.
C DIE ROUTINEN Q1...QI SIND DIE ENTSPRECHENDEN VERSCHIEBUNGSANSAETZE
C ALS FUNKTION DES ORTES(X) UND MUESSEN DEM PROGRAMM PROBLEMSPEZIFISCH
C ANGEPASST UEBERSETZT UND GELINKT WERDEN. DIE ROUTINE Q MUSS
C ENTSPRECHEND MODIFIZIERT WERDEN
C LETZTE AENDERUNG 5.4.84
  PARAMETER (IFELD=300)
  PARAMETER (IANS=3)
  PARAMETER (INULL=3*IFELD)
  PARAMETER (INNUL=6*IFELD)
  REAL PK(2,IFELD),GV(2,IFELD/2),POT(IFELD),XPOL(2),HM(IANS,IANS)
  INTEGER KN(8,IFELD/2),BODKN(IFELD/2),SIGKN(IFELD/2),ZEIGER(IFELD)
  DATA KN,ZEIGER,BODKN,SIGKN/INNUL*0/
  DATA PK,POT/INULL*0.0/
C COMMONBLOCK ZUR UEBERGABE DER KOORDINATEN DES ROTATIONSPOLS AN DIE
C ROTARIONSANSATZFUNKTION Q3
  COMMON/POL/XPOL
C INITIALISIEREN DER MASSENMATRIX
  CALL NULLMACH(HM,IANS**2)
C LESEN DER NETZDATEN
  CALL LESEN(5,NK,PK,NE,KN,ZEIGER,BODKN,SIGKN,GV,NKS,XPOL)
C BERECHNUNGEN FUER ALLE ANSATZFUNKTIONEN
  DO 10 I=1,IANS
C BESETZEN DES GESCHWINDIGKEITSVEKTORS
    DO 1 J=1,NK
      IF(BODKN(J).EQ.0) GOTO 2
1      CALL Q(I,GV(1,J),PK(1,BODKN(J)))
C BERECHNUNG DES POTENTIALS
2      CALL POT2D(PK,KN,NK,NE,ZEIGER,BODKN,SIGKN,NKS,GV,POT,6,D)
C INTEGRATION DER GEWICHTETEN DRUECKE
      CALL PINT(HM(1,I),POT,PK,KN,NE,ZEIGER)
C NEUINITIALISIERUNG DES VEKTORS POT
      CALL NULLMACH(POT,NK)
10     CONTINUE
      CALL MAPRINT(HM,IANS,IANS,IANS,6)
      STOP
      END

SUBROUTINE Q(I,RES,X)
C UNTERPROGRAMM ZUR BERECHNUNG DES I-TEN VERSCHIEBUNGSANSATZES
C RES(1),RES(2) ALS FUNKTION VON X(1),X(2)

```

C DIE ANSATZFUNKTIONEN Q1 MUESSEN PROBLEMSPEZIFISCH DAZUGELADEN WERDEN  
 C UND DIE GOTO ANWEISUNG MUSS ENTSPRECHEND DER ANZAHL DER ANSATZFUNKTIONEN  
 C GEFUELLT SEIN.

C LETZTE AENDERUNG 29.2.84

```

    REAL RES(2),X(2)
    GOTO (1,2,3),I
    RETURN
1    CALL Q1(RES,X)
    RETURN
2    CALL Q2(RES,X)
    RETURN
3    CALL Q3(RES,X)
    RETURN
    END
  
```

SUBROUTINE Q1(RES,X)

C VERSCHIEBUNGSANSATZ EINHEITSVERSCHIEBUNG IN X-RICHTUNG

C LETZTE AENDERUNG 29.2.84

```

    REAL RES(2),X(2)
    RES(1)=1.0
    RES(2)=0.0
    RETURN
    END
  
```

SUBROUTINE Q2(RES,X)

C VERSCHIEBUNGSANSATZ EINHEITSVERSCHIEBUNG IN Y-RICHTUNG

C LETZTE AENDERUNG 29.2.84

```

    REAL RES(2),X(2)
    RES(1)=0.0
    RES(2)=1.0
    RETURN
    END
  
```

SUBROUTINE Q3(RES,X)

C VERSCHIEBUNGSANSATZ FUER EINHEITSDREHUNG UM XO

C LETZTE AENDERUNG 29.2.84

```

    REAL RES(2),X(2),XO(2),W(2)
    COMMON /POL/XO
    CALL VEKADD(W,X,XO,-1.,2)
    CALL ORTVEK(RES,W)
    RETURN
    END
  
```

SUBROUTINE PINT(HM,POT,PK,KN,NE,ZEIGER)

C UNTERPROGRAMM BILDET INTEGRAL POT\*Q ds FUER ALLE Q

C LETZTE AENDERUNG 1.3.84

```

    PARAMETER (IANS=3)
    REAL POT(1),HM(1),X(2,8),DM(IANS),PK(2,1),ELPOT(8)
    INTEGER KN(8,1),ZEIGER(1),KK(8)
    DO 100 I=1,NE
    IF (ZEIGER(KN(3,I))).LE.D) GOTO 100
    IF (ZEIGER(KN(4,I))).LE.D) GOTO 100
    CALL NULLMACH(ELPOT,8)
  
```

```

DO 10 J=1,8
KK(J)=KN(J,I)
IF (KK(J).EQ.0) GOTO 10
ELPOT(J)=POT(KK(J))
X(1,J)=PK(1,KK(J))
X(2,J)=PK(2,KK(J))
10  CONTINUE
CALL NULLMACH(DM,IANS)
CALL ELEPINT(KK,X,ELPOT,ZEIGER,IANS,DM,1)
IF (ZEIGER(KK(1)).GT.0)CALL ELEPINT(KK,X,ELPOT,ZEIGER,IANS,DM,1)
DO 20 J=1,IANS
20  HM(J)=HM(J)+DM(J)
100  CONTINUE
RETURN
END

```

```

SUBROUTINE ELEPINT(KK,X,ELPOT,ZEIGER,IANS,DM,IRL)
C UNTERPROGRAMM BILDET DIE INTEGRALE FUER PINT ENTLANG EINES ELEMENTRANDES
C LETZTE AENDERUNG 1.3.84
REAL X(2,8),RES(2),DM(1),NV(2),R(2),JM(2,2),XP(2),H(8),DHD(2,8)
REAL ELPOT(8)
INTEGER KK(8)
COMMON /GAUSS/ STELLE(5),GEW(5),NSTUETZ
R(1)=1.
R(2)=-1.
C INTEGRATION NACH GAUSS
DO 100 I=1,NSTUETZ
R(IRL)=STELLE(I)*2.-1.
FAK=GEW(I)*2.
CALL ANSATZ(R,KK,H,DHD)
CALL JACOBI(X,DHD,JM)
C BERECHNUNG DES NORMALENVEKTORS UND SEINES BETRAGS
CALL NVEKT(NV,JM,IRL)
CALL VBET(VABS,NV,2)
C BERECHNUNG DER LAENGE DES BOGENELEMENTS
CALL DBOG(DS,JM,IRL)
C BERECHNUNG DER GLOBALEN KOORDINATEN
CALL XINTER(XP,X,H)
C BERECHNUNG DES POTEBTIALS AN DER STELLE XP
CALL POTINTER(XPOT,ELPOT,H)
C AUFSUMMIERUNG DER INTEGRALE FUER ALLE ANSATZFUNKTIONEN
DO 100 J=1,IANS
CALL Q(J,RES,XP)
100  DM(J)=DM(J)+XPOT*SKALP(RES,NV,2)/VABS*DS*FAK
RETURN
END

```

```

SUBROUTINE POTINTER(XPOT,ELPOT,H)
C UNTERPROGRAMM ZUR INTERPOLATION FUER 4-8 KNOTIGE ISOPARAMETRISCHE
C ELEMENTE
C LETZTE AENDERUNG 1.3.84
REAL ELPOT(8),H(8)
XPOT=0.
DO 1 I=1,8
1  XPOT=XPOT+ELPOT(I)*H(I)

```

RETURN  
END

```

SUBROUTINE RASTGEN(PK,KN,ZEIGER,BODKN,SIGKN,NE,NK,NKB,NKS,IED,IPD)
C UNTERPROGRAMM ZUR GENERIERUNG VON DATEN FUER DAS PROGRAMM POT2D
C AUS DER NETZGEOMETRIE
C LETZTE AENDERUNG 22.2.84
C IST DER PARAMETER IPD>0 WIRD EIN PLOT DER EINGABEDATEN AUF DER DATEI IPD
C ERSTELLT
C DIE DATEI IED MUSS FOLGENDE DATEN ENTHALTEN:
C "NK" ANZAHL DER KNOTEN
C NK-MAL:
C "X-KOORDINATE" "Y-KOORDINATE"
C "NE" ANZAHL DER ELEMENTE
C NE-MAL
C "ELEMENTKNOTENNUMMERN 1...8"
C DIE ORIENTIERUNG DER ELEMENTE IST BELIEBIG
C "NKB" ANZAHL DER KNOTEN AUF DER KOERPEROBERFLAECHE
C NKB-MAL
C "NUMMER EINES KOERPERKNOTEN"
C "NKS" ANZAHL DER KNOTEN AUF DEM AUSSENRAND
C NKS-MAL
C "NUMMER EINES AUSSENKNOTEN"
C "NKD" ANZAHL DER DUMMYKNOTEN
C NKD-MAL
C "NUMMER EINES DUMMYKNOTEN"
REAL PK(2,1)
INTEGER KN(8,1),ZEIGER(1),BODKN(1),SIGKN(1),DUMMKN(50)
CALL LESRAST(IED,NK,PK,NE,KN,NKS,SIGKN,NKB,BODKN,NKD,DUMMKN)
WRITE(6,('NE=',I4,'NK=',I4,'NKB=',I4,'NKS=',I4,
1'NKD=',I4)')NE,NK,NKB,NKS,NKD
C PLOTTEN DER EINGABEDATEN
IF (IPD.GT.0)
A CALL RASTPLOT(KN,PK,BODKN,SIGKN,DUMMKN,NE,NK,NKB,NKS,NKD,IPD)
C BESEITIGEN DER DUMMYKNOTEN
CALL DUMMKILL(KN,BODKN,SIGKN,PK,NE,NKB,NKS,NK,DUMMKN,NKD)
C UMSORTIEREN, SO, DASS DIE LETZTEN NKS KNOTEN DIE AUSSENRANDKNOTEN SIND
CALL IREIHE(SIGKN,NKS)
DO 10 I=NKS,1,-1
IF (SIGKN(I).LT.NK-NKS+1) THEN
CALL KNSW(PK,KN,SIGKN,BODKN,NE,NK,NK-NKS+1,SIGKN(1))
IZ=SIGKN(1)
DO 5 J=1,I
SIGKN(J)=SIGKN(J+1)
5 SIGKN(I)=IZ
END IF
10 CONTINUE
C BESETZUNG DES FELDES ZEIGER
DO 20 I=1,NKB
20 ZEIGER(BODKN(I))=I
DO 30 I=1,NKS
30 ZEIGER(SIGKN(I))=-I
C SPIEGELUNG VON RECHTSHERUM NUMMERIERTEN ELEMENTEN
CALL SORT(PK,KN,NE)
C DREHUNG DER RANDELEMENTE DAMIT DIE ERWARTETEN KANTEN AN DER
C KOERPEROBERFLAECHE LIEGEN

```

```

CALL ORIENT(KN, ZEIGER, NE)
RETURN
END

```

```

SUBROUTINE UMGEN(PK, KN, ZEIGER, BODKN, SIGKN, NE, NK, NKB, NKS, XPOL, IED)
C UNTERPROGRAM ZUR GENERIERUNG VON DATENFILES FUER DAS PROGRAM POT2D
C LETZTE AENDERUNG 20.2.84
C DIE EINGABEDATEI IED SOLL ENTHALTEN
C "NKB"      =ANZAHL DER KNOTEN AUF DEM UMFANG
C NKB MAL:
C      "KNOTENUMMER" "X-KOORDINATE" "Y-KOORDINATE"
C "NEU"      =ANZAHL DER ELEMENTE AUF DEM UMFANG
C NEU MAL:
C      "NUMMER DES ELEMENTS"
C      DIE ELEMENTE SOLLEN IM GEGENUHRZEIGERSINN
C      IN AUFSTIEGENDER REIHENFOLGE NUMMERIERT SEIN
C      "NUMMER DES ERSTEN ELEMNTKNOTEN AUF DEN MAN BEIM
C      UMFahren DES KOEPERS GEGEN DEN UHRZEIGERSINN
C      TRIFFT"
C      "NUMMER DES MITTLEREN ELEMENT KNOTEN"(DWENN NICHT
C      VORHANDEN)
C      "NUMMER DES LETZTEN ELEMNTKNOTEN AUF DEN MAN BEIM
C      UMFahren DES KOEPERS GEGEN DEN UHRZEIGERSINN
C      TRIFFT"
C "KS"      0 WENN KEINE SEITLICHEN ZWISCHENKNOTEN GENERIERT WERDEN
C      SOLLEN
C      1 WENN SEITLICHE ZWISCHENKNOTEN GENERIERT WERDEN SOLLEN
C "KA"      0 WENN KEINE AUSSEREN ZWISCHENKNOTEN GENERIERT WERDEN
C      SOLLEN
C      1 WENN AUSSERE ZWISCHENKNOTEN GENERIERT WERDEN SOLLEN
C "XPOL" X,Y-KOORDINATE DES POLS FUER ETWAIGE DREHUNGEN
REAL PK(2,1), XPOL(2)
INTEGER KN(8,1), ZEIGER(1), BODKN(1), SIGKN(1)
CALL LESUM(IED, NKB, PK, NEU, KN, KS, KA, XPOL)
NK=NKB
NE=NEU
CALL NUMGEN(NK, NE, NEU, KN, KS, KA)
CALL KOOGEN(NE, NEU, KN, NK, PK)
CALL ZEIGUM(ZEIGER, BODKN, SIGKN, NE, NEU, KN, NKB, NKS)
C DREHUNG DER RANDELEMENTE DAMIT DIE ERWARTETEN KANTEN AN DER
C KOEPEROBERFLAECHE LIEGEN
CALL ORIENT(KN, ZEIGER, NE)
RETURN
END

```

```

SUBROUTINE KOOGEN(NE, NEU, KN, NK, PK)
C UNTERPROGRAM ZUR GENERIERUNG DER KNOTENKOORDINATEN
C LETZTE AENDERUNG 17.1.84
INTEGER KN(8,1)
REAL PK(2,1), V(2), W(2), XK(2,0:3)
NEA=NE-NEU+1
C GENERIERUNG DER KOORDINATEN DER AUSSENECKEN
DO 100 I=NEA, NE-1
CALL VEKADD(V, PK(1, KN(4, I+1)), PK(1, KN(1, I)), -1., 2)
CALL ORTVEK(V, V)

```

```

100 CALL VEKADD(PK(1,KN(3,I)),PK(1,KN(4,I)),V,-0.6,2)
    CALL VEKADD(V,PK(1,KN(4,NEA)),PK(1,KN(1,NE)), -1.,2)
    CALL ORTVEK(V,V)
    CALL VEKADD(PK(1,KN(3,NE)),PK(1,KN(4,NE)),V,-0.6,2)
    DO 200 I=NEA,NE
    IF (KN(5,I).EQ.0) GOTO 150
C GENERIEREN DER SEITLICHEN ZWISCHENKNOTEN
    CALL VEKADD(V,PK(1,KN(2,I)),PK(1,KN(1,I)), -1.,2)
    CALL VEKADD(PK(1,KN(5,I)),PK(1,KN(1,I)),V,0.5,2)
150 IF (KN(6,I).EQ.0) GOTO 200
C GENERIEREN DER AUESSEREN ZWISCHENKNOTEN
    IF (I.EQ.NEA) THEN
        NKV=KN(2,NE)
    ELSE
        NKV=KN(2,I-1)
    END IF
    IF (I.EQ.NE) THEN
        NKN=KN(3,NEA)
    ELSE
        NKN=KN(3,I+1)
    END IF
    DO 160 K=1,2
    XK(K,0)=PK(K,NKV)
    XK(K,1)=PK(K,KN(2,I))
    XK(K,2)=PK(K,KN(3,I))
160 XK(K,3)=PK(K,NKN)
    CALL ISOINT(V,XK(1,0),-0.5)
    CALL ISOINT(W,XK(1,1),0.5)
    CALL VEKADD(PK(1,KN(6,I)),V,W,0.5,2)
    CALL VEKADD(PK(1,KN(6,I)),PK(1,KN(6,I)),V,-0.5,2)
200 CONTINUE
    RETURN
    END

```

SUBROUTINE RASTPLOT

```

A (KN,PK,BODKN,SIGKN,DUMMKN,NE,NK,NKB,NKS,NKD,IPD)
C MALT DAS EINGELESENE EINGELESENE RASTER UND KENNZEICHNET ALLE KOERPER-,
C AUSSEN- UND DUMMYKNOTEN
C PLOTDATEI IST IPD
C LETZTE AENDERUNG 22.2.84
    REAL PK(2,1),XD(2)
    INTEGER KN(8,1),BODKN(1),SIGKN(1),DUMMKN(1)
C BESTIMMUNG DES MASSTABS UND DES KOORDINATENURSPRUNGS
    CALL DINA4(XD(1),XD(2),PK,NK,SCAL)
C EROEFFNUNG DER PLOTDATEI
    CALL PLOTS(0,0,IPD)
C DEFINITION DES URSPRUNGS
    CALL PLOT(XD(1),XD(2),-3)
C PLOTTEN DES NETZES
    CALL NETZPLO(KN,PK,NK,NE,SCAL)
    CALL ELENUM(KN,PK,NK,NE,SCAL)
C KENNZEICHNEN DER KOERPERKNOTEN
    CALL KENNZPLO(PK,BODKN,NKB,SCAL,2)
C KENNZEICHNEN DER AUSSENKNOTEN
    CALL KENNZPLO(PK,SIGKN,NKS,SCAL,5)
C KENNZEICHNEN DER DUMMYKNOTEN

```

```

      CALL KENNZPLO(PK,DUMMKN,NKD,SCAL,D)
C SCHLIESSEN DER PLOTDATEI
      CALL PLOT(D,D,999)
      RETURN
      END

```

```

      SUBROUTINE DATWRIT
      1 (IDAT,NK,PK,NE,KN,ZEIGER,BODKN,SIGKN,NKB,NKS,GV,XPOL)
C UNTERPROGRAMM ZUM SCHREIBEN DER DATEN FUER POT2D AUF IDAT
C LETZTE AENDERUNG 28.2.84
      COMMON/GESCHW/V,PHIV
      REAL PK(2,1),GV(2,1),XPOL(2)
      INTEGER ZEIGER(1),BODKN(1),SIGKN(1),KN(8,1)
      WRITE(IDAT,*)NK
      WRITE(IDAT,*)((PK(I,J),I=1,2),J=1,NK)
      WRITE(IDAT,*)NE
      WRITE(IDAT,*)((KN(I,J),I=1,8),J=1,NE)
      WRITE(IDAT,*)(ZEIGER(I),I=1,NK)
      WRITE(IDAT,*)NKB
      WRITE(IDAT,*)(BODKN(I),I=1,NKB)
      WRITE(IDAT,*)NKS
      WRITE(IDAT,*)(SIGKN(I),I=1,NKS)
      WRITE(IDAT,*)(GV(1,I),GV(2,I),I=1,NKB)
      WRITE(IDAT,*)XPOL(1),XPOL(2)
      RETURN
      END

```

```

      SUBROUTINE EINDRUCK(IDAT,PK,KN,NK,NE,ZEIGER,GV)
C UNTERPROGRAMM ZUM AUSDRUCK DER ELEMENTGEOMETRIE AUF DIE DATEI IDAT
C LETZTE AENDERUNG 1.3.84
      REAL PK(2,1),GV(2,1)
      INTEGER KN(8,1),ZEIGER(1)
      WRITE(IDAT,21)NK,NE
      DO 10 I=1,NE
        DO 10 J=1,8
          IF (KN(J,I).GT.0) THEN
            IF (ZEIGER(KN(J,I)).GT.0)WRITE(IDAT,3) GV(1,ZEIGER(KN(J,I)))
            1,GV(2,ZEIGER(KN(J,I)))
            WRITE(IDAT,2)I,J,KN(J,I),ZEIGER(KN(J,I)),PK(1,KN(J,I)),
            1 PK(2,KN(J,I))
          END IF
        10 CONTINUE
      RETURN
      1 FORMAT(2I3,1F16.8)
      21 FORMAT(' ANZAHL DER KNOTEN: ',I5,
      &' ANZAHL DER ELEMENTE: ',I5,
      &' ELNR LKNNR GKNNR ZEIGER X Y VX VY ',/)
      2 FORMAT('+',4I5,2F10.5,/)
      3 FORMAT('+',40X,2F10.5)
      4 FORMAT(I5,2X,E16.8)
      END

```

```

      SUBROUTINE LESEN(IDAT,NK,PK,NE,KN,ZEIGER,BODKN,SIGKN,GV,NKS,XPOL)
C UNTERPROGRAMM ZUM LESEN DER DATEN FUER POT2D VON IDAT

```

C LETZTE AENDERUNG 27.2.84

```

REAL PK(2,1),GV(2,1),XPOL(2)
INTEGER ZEIGER(1),BODKN(1),SIGKN(1),KN(8,1)
READ(IDAT,*)NK
READ(IDAT,*)((PK(I,J),I=1,2),J=1,NK)
READ(IDAT,*)NE
READ(IDAT,*)((KN(I,J),I=1,8),J=1,NE)
READ(IDAT,*)(ZEIGER(I),I=1,NK)
READ(IDAT,*)NKB
READ(IDAT,*)(BODKN(I),I=1,NKB)
READ(IDAT,*)NKS
READ(IDAT,*)(SIGKN(I),I=1,NKS)
READ(IDAT,*)(GV(1,I),GV(2,I),I=1,NKB)
READ(IDAT,*)XPOL(1),XPOL(2)
RETURN
END

```

SUBROUTINE DIGEN

C PROGRAM ERZEUGT AUS DEN MIT DIGIP AUFGEMESSENEN KOORDINATEN

C EINGABEDATEN FUER DIE NETZGENERIERUNG

C LETZTE AENDERUNG 2.4.84

```

CHARACTER EINFIL*80,JANE*2
PRINT*, 'Dieses Unterprogramm erzeugt aus den mit "DIGIP"
1 digitalisierten Daten'
PRINT*, 'Eingabedaten fuer die Netzgenerierung'
PRINT*, '
WRITE(*,(' Auf welcher Datei sind die Daten',*))
READ(*,('A')) EINFIL
OPEN(2,FILE=EINFIL,STATUS='UNKNOWN')
OPEN(7,FILE='DUM.DAT',STATUS='SCRATCH')
READ(2,*)NKU
NKU=NKU-1
WRITE(7,*)NKU
READ(2,*) J,X0,Y0
WRITE(7,*) J,D.,D.
DO 10 I=2,NKU
READ(2,*)J,X,Y
10 WRITE(7,*)J,(X-X0)/400.,(Y-Y0)/400.
READ(2,*)J,XPOL,YPOL
15 WRITE(6,(' Sollen die Elemente auf der Koerperoberflaeche
1 Zwischenknoten habenJa/Ne ',*))
READ(*,('A'))JANE
CALL GROSSBU(JANE,2)
IF (JANE.EQ.'JA') THEN
NE=(NKU+1)/2
WRITE(7,*) NE
WRITE(7,*) (I, 2*I-1, 2*I, 2*I+1, I=1,NE-1)
IF (2*NE.EQ.NKU) THEN
WRITE(7,*) NE, NKU-1, NKU, 1
ELSE
WRITE(7,*) NE, NKU, 0, 1
END IF
GOTO 20
END IF
IF (JANE.EQ.'NE') THEN
NE=NKU

```

```

        WRITE(7,*) NE
        WRITE(7,*) (I,I,0,I+1,I=1,NE-1)
        WRITE(7,*) NE,NKU,0,1
    ELSE
        GOTO 15
    END IF
20    WRITE(6,('' Sollen seitliche Zwischenknoten generiert werden
1 Ja/Ne '',,$)')
    READ(*,'(A)') JANE
    CALL GROSSBU(JANE,2)
    IF (JANE.EQ.'NE') THEN
        KS=0
    ELSE
        IF (JANE.NE.'JA') GO TO 20
        KS=1
    END IF
30    WRITE(6,('' Sollen aeussere Zwischenknoten generiert werden
1 Ja/Ne '',,$)')
    READ(*,'(A)') JANE
    CALL GROSSBU(JANE,2)
    IF (JANE.EQ.'NE') THEN
        KA=0
    ELSE
        IF (JANE.NE.'JA') GO TO 20
        KA=1
    END IF
    WRITE(7,*)KS,KA
    WRITE(7,*)(XPOL-X0)/400.,(YPOL-Y0)/400.
    RETURN
    END

```

## PROGRAM INTAKTPOT

```

C INTERAKTIVES PROGRAMM ZUR STEUERUNG VON POT2D UND DEN ZUGEHORIGEN
C GENERIERUNGSPROGRAMMEN UMGEN UND RASTGEN, SOWIE ZUM SPEICHERN VON
C NETZDATEN MIT DEM UNTERPROGRAMM SCHREIBEN
C DER PARAMETER IFELD SOLL MINDESTENS SO GRAOSS SEIN WIE DIE ANZAHL
C ALLER KNOTEN IM NETZ UND MUSS IM UNTERPROGRAMM POT2D AUF DIE GLEICHE
C GROESSE GESETZT SEIN.
C LETZTE AENDERUNG 2.4.84
    CHARACTER JANE*2,NFILE*25
    PARAMETER(IFELD=300)
    PARAMETER(INULL=3*IFELD+2*(IFELD/2))
    PARAMETER(INNUL=6*IFELD)
    REAL PK(2,IFELD),POT(IFELD),GV(2,IFELD/2),GSTAT(2),XPOL(2)
    INTEGER ZEIGER(IFELD),BODKN(IFELD/2),SIGKN(IFELD/2),KN(8,IFELD/2)
    INTEGER KPR,RPR
    DATA KN,ZEIGER,BODKN,SIGKN / INNUL*0 /
    DATA POT,PK,GV /INULL*0.0/
    CALL CPUTIM(CP,CT)
    WRITE(6,('' Programm zur Berechnung von Potentialstroemungen
1'',/))
11    WRITE(6,('' Sollen die Netzdaten aus digitalisierten Daten
1 Der Knoten an der''/
2 '' Koerperoberflaeche erzeugt werdenJa/Ne '',,$)')
    READ(*,'(A)')JANE
    CALL GROSSBU(JANE,2)

```

```

IF (JANE.EQ.'JA') THEN
    CALL DIGEN
    REWIND 7
    CALL UMGEN(PK,KN,ZEIGER,BODKN,SIGKN,NE,NK,NKB,NKS,XPOL,7)
    GOTO 14
ELSE
    IF (JANE.NE.'NE') GOTO 11
END IF
1   WRITE(6,('' Name der Eingabedatei'', $)')
    READ(*, '(A)')NFILE
    OPEN(7, FILE=NFILE, STATUS='UNKNOWN')
12  WRITE(6,('' Soll das Netz aus einem Raster generiert werden
1 Ja/Ne '', $)')
    READ(*, '(A)')JANE
    CALL GROSSBU(JANE,2)
    IF (JANE.EQ.'JA') THEN
121  WRITE(6,('' Soll das Raster geplottet werdenJa/Ne '',
1 $)')
        READ(*, '(A)')JANE
        CALL GROSSBU(JANE,2)
        IF (JANE.EQ.'JA') THEN
            WRITE(6,('' Nummer der Plotdatei'', $)')
            READ*, IPD
        ELSE
            IF (JANE.NE.'NE') GOTO 12
            IPD=0
        END IF
        CALL RASTGEN(PK,KN,ZEIGER,BODKN,SIGKN,NE,NK,NKB,NKS,7,IPD)
        GOTO 14
    ELSE
        IF (JANE.NE.'NE') GOTO 12
    END IF
13  WRITE(6,('' Sind die Eingabedaten bereits vollstaendig
1 vorhandenJa/Ne '', $)')
    READ(*, '(A)')JANE
    CALL GROSSBU(JANE,2)
    IF (JANE.EQ.'JA') THEN
        CALL LESEN(7,NK,PK,NE,KN,ZEIGER,BODKN,SIGKN,GV,NKS,XPOL)
        GOTO 14
    ELSE
        IF (JANE.NE.'NE') GOTO 11
        GOTO 1
    END IF
14  PRINT*, 'Im folgenden brauchen nur Zahlen eingegeben zu werden
1 wo Abweichungen von '
    PRINT*, 'den angegebenen gewuenscht werden, sonst <RET>'
141  WRITE(6,('' X-Koordinate der Rotationsachse='',F6.3, ''',
1 $)')XPOL(1)
    CALL GELES(5,XPOL(1),*141)
142  WRITE(6,('' Y-Koordinate der Rotationsachse='',F6.3, ''',
1 $)')XPOL(2)
    CALL GELES(5,XPOL(2),*142)
143  WRITE(6,('' Winkelgeschwindigkeit (Bogenmass)='',F6.3,
1 ''', $)')PHIPU
    CALL GELES(5,PHIPU,*143)
144  WRITE(6,('' X-Komponente der Anstroemgeschwindigkeit=''
1,F6.3, ''', $)')GSTAT(1)

```

```

      CALL GELES(5,GSTAT(1),*144)
145  WRITE(6,(' Y-Komponente der Anstroemgeschwindigkeit='
1,F6.3,(',$)')GSTAT(2)
      CALL GELES(5,GSTAT(2),*145)
      CALL STARKGV(GV,GSTAT,XPOL,PHIPU,PK,BODKN,NK)
15  WRITE(6,(' Sollen die Netzdaten gespeichert werdenJa/Ne '
1,$)')
      READ(*,'(A)')JANE
      CALL GROSSBU(JANE,2)
      IF (JANE.EQ.'JA') THEN
        WRITE(6,(' Dateiname',$)')
        READ(*,'(A)')NFILE
        OPEN(9,FILE=NFILE,STATUS='UNKNOWN')
        CALL DATWRIT(9,NK,PK,NE,KN,ZEIGER,BODKN,SIGKN,NKB,NKS,GV,XPOL)
      ELSE
        IF (JANE.NE.'NE') GOTO 15
      END IF
2  CALL CPUTIM(CP,CT)
      CALL TEXPRI('Fuer das Lesen und Generieren',6)
      WRITE(6,1000)CP
21  WRITE(6,(' Erfolgt die Ausgabe auf den BildschirmJa/Ne ',$)')
      READ(*,'(A)')JANE
      CALL GROSSBU(JANE,2)
      IF (JANE.EQ.'NE') THEN
        WRITE(6,(' Name der Ausgabedatei',$)')
        READ(*,'(A)')NFILE
        OPEN(8,FILE=NFILE,STATUS='UNKNOWN')
        IDAT=8
      ELSE
        IF (JANE.NE.'JA') GO TO 21
        IDAT=6
      END IF
3  WRITE(6,(' Sollen die Eingabedaten ausgegeben werdenJa/Ne '
1,$)')
      READ(*,'(A)')JANE
      CALL GROSSBU(JANE,2)
      IF (JANE.EQ.'JA') THEN
        CALL EINDRUCK(IDAT,PK,KN,NK,NE,ZEIGER,GV)
        CALL CPUTIM(CP,CT)
        CALL TEXPRI('Fuer das Schreiben der Eingabedaten',6)
        WRITE(6,1000)CP
      ELSE
        IF (JANE.NE.'NE') GO TO 3
      END IF
4  WRITE(6,(' Soll das Gleichungssystem ausgegeben werdenJa/Ne '
1,$)')
      READ(*,'(A)')JANE
      CALL GROSSBU(JANE,2)
      IF (JANE.EQ.'NE') THEN
        KPR=0
      ELSE
        KPR=1
        IF (JANE.NE.'JA') GO TO 4
      END IF
*****
      CALL POT2D(PK,KN,NK,NE,ZEIGER,BODKN,SIGKN,NKS,GV,
1 POT,IDAT,KPR)

```

```

C*****
5  WRITE(6,(' SoLlen die Potentiale ausgegeben werdenJa/Ne '
1,$))
   READ(*,(A))JANE
   CALL GROSSBU(JANE,2)
   IF (JANE.EQ.'JA') THEN
       WRITE(6,(' Falls eine andere Ausgabedatei bitte
1Dateiname eingeben, sonst <RET> ',,$))
       READ(*,(A))NFILE
       IF (NFILE.NE.' ') THEN
           OPEN(8,FILE=NFILE,STATUS='UNKNOWN')
           IDAT=8
       END IF
       CALL POTPRINT(IDAT,POT,NK)
   ELSE
       IF (JANE.NE.'NE') GO TO 5
   END IF
6  WRITE(6,(' SoLl das Netz und das Geschwindigkeitsfeld
1 geplottet werdenJa/Ne ',,$))
   READ(*,(A))JANE
   CALL GROSSBU(JANE,2)
   IF (JANE.EQ.'JA') THEN
       WRITE(6,(' Nummer der Plotdatei',,$))
       READ*,IPD
       CALL CPUTIM(CP,CT)
       CALL MALEN(KN,PK,NK,NE,ZEIGER,POT,GSTAT,XPOL,.5,1,IPD)
       CALL CPUTIM(CP,CT)
       CALL TEXPRI('Fuer das Erstellen der Plotdatei',6)
       WRITE(6,1000)CP
   ELSE
       IF (JANE.NE.'NE') GO TO 6
   END IF
   CALL CPUTIM(CP,CT)
   WRITE(6,(' CPU-Zeit gesamt:',F7.3,' SEC'))CT
1000 FORMAT(' verbrauchte CPU-Zeit:',F7.3,'SEC')
   STOP
   END

```

```

SUBROUTINE STARKGV(GV,GSTAT,XPOL,PHIPU,PK,BODKN,NK)
C UNTERPROGRAMM ZUR BESETZUNG DES VEKTORS GV DER KOERPERGESCHWINDIGKEITEN
C MIT DEM ANTEIL AUS DER STARRKOERPERBEWEGUNG
C DER VEKTOR GV MUSS BEREITS MIT DEFINIERTEN WERTEN BESETZT SEIN (Z.B. 0.0)
C LETZTE AENDERUNG 5.4.84
   REAL GV(2,1),PK(2,1),XPOL(2),GSTAT(2),W(2)
   INTEGER BODKN(1)
   DO 10 I=1,NK
       II=BODKN(I)
       IF (II.EQ.0) RETURN
       CALL VEKADD(W,PK(1,II),XPOL,-1.,2)
       CALL ORTVEK(W,W)
   DO 10 J=1,2
10  GV(J,I)=GSTAT(J)+W(J)*PHIPU+GV(J,I)
   RETURN
   END

```

```
      SUBROUTINE POTPRINT(IDAT,POT,NK)
C UNTERPROGRAMM SCHREIBT POTENTIALE ALLER KNOTEN AUF IDAT
C LETZTE AENDERUNG 29.2.84
      REAL POT(1)
      WRITE(IDAT,'(3(I4,F16.8))') (I,POT(I),I=1,NK)
      RETURN
      END
```