KFKI
29/1969

NUMERICAL CALCULATIONS FOR THE EVALUATION OF MÖSSBAUER SPECTRA

L. Cser and I. Gladkih

HUNGARIAN ACADEMY OF SCIENCES
CENTRAL RESEARCH INSTITUTE FOR PHYSICS

BUDAPEST

NUMERICAL CALCULATIONS FOR THE EVALUATION OF MÖSSBAUER SPECTRA

L. Cser, I. Gladkih

Central Research Institute for Physics, Budapest, Hungary

1.§. The Mössbauer effect is most frequently observed on the measured absorption spectra . The Mössbauer parameters of the measured spectra are the position, width and shape of the absorption lines, the maximum absorption /amplitude/ and the base line background.

For thin absorber the absorption lines are shaped like Lorentzian curves. All the lines in the spectra considered in the following are assumed to have Lorentzian shape. The positions and intensities of the observed lines are determined by the eigenvalues and eigenvectors of the hyperfine interaction Hamiltonian of the nuclear levels, respectively.

The hyperfine interaction Hamiltonian is given by the sum (1) [1]

$$\hat{H} = \hat{H}_Q + H_M \qquad (1)$$

If the coordinate system is chosen to be such that the z-axis points in the direction of the component V_{zz} of the electric field gradient /Fig.1/, then the contribution from the quadrupole interactions is defined as

$$\hat{H}_Q = \frac{eQV_{zz}}{4I(2I-1)}\left[3I_z^2 - I(I+1) + \eta\left(I_x^2 - I_y^2\right)\right]$$

and the contribution from the magnetic interactions is defined as

$$\hat{H}_M = -g\mu_N H_i\left[I_z\cos\theta + \left(I_x\cdot\cos\varphi + I_y\sin\varphi\right)\cdot\sin\theta\right]$$

where

Q = the electric quadrupole momentum

e = the proton charge

g = the nuclear magnetic g-factor

I = the nuclear spin

μ_N = the nuclear magneton

$\eta = \dfrac{V_{xx} - V_{yy}}{V_{zz}}$ = the asymmetry parameter

θ, φ = the polar coordinates of the magnetic field H_i relative to the principal axes of the electric field gradient tensor

I_x, I_y, I_z are the angular momentum operators. The components of the electric field gradient are chosen to be such that we have $|V_{zz}| \geq |V_{yy}| \geq |V_{xx}|$, hence $0 \leq \eta \leq 1$

For $I = 1/2$ and $I = 3/2$ e.g. ^{57}Fe and ^{119}Sn the matrix elements of the Hamiltonian defined by /1/ can be written in the form [2]

m'	m: 1/2	m: - 1/2
1/2	b_{11}	b_{12}
-1/2	b_{21}	b_{22}

(I)

m'	m				
	$3/2$	$1/2$	$-1/2$	$-3/2$	
$3/2$	a_{11}	a_{12}	a_{13}	0	(II)
$1/2$	a_{21}	a_{22}	a_{23}	a_{13}	
$-1/2$	a_{13}	a_{32}	a_{33}	a_{12}	
$-3/2$	0	a_{13}	a_{21}	a_{44}	

where

$$b_{11} = -1/2\,\beta\cos\theta$$
$$b_{22} = 1/2\,\beta\cos\theta$$
$$b_{12} = -1/2\,\beta\sin\theta\exp(-i\varphi)$$
$$b_{21} = -1/2\,\beta\sin\theta\exp(i\varphi)$$
$$a_{11} = 3A - 3/2\,\alpha\cos\theta$$
$$a_{22} = -3A - 1/2\,\alpha\cos\theta$$
$$a_{33} = -3A + 1/2\,\alpha\cos\theta$$
$$a_{44} = 3A + 3/2\,\alpha\cos\theta$$
$$a_{12} = -\sqrt{3}/2\,\alpha\sin\theta\exp(-i\varphi)$$
$$a_{21} = -\sqrt{3}/2\,\alpha\sin\theta\exp(i\varphi)$$
$$a_{23} = -\alpha\cdot\sin\theta\exp(-i\varphi)$$
$$a_{32} = -\alpha\cdot\sin\theta\exp(i\varphi)$$
$$a_{13} = \sqrt{3}\,\eta A$$
$$\alpha = g_{3/2}\cdot\mu_N\cdot H_i$$
$$\beta = g_{1/2}\cdot\mu_N\cdot H_i$$
$$A = eQV_{zz}/4I(2I-1)$$

The positions /TE/ of the absorption lines are given in terms of the eigenvalues of the above matrices, by

$$TE(3/2,i;\,1/2,j) = \lambda(3/2,i) - \lambda(1/2,j) \qquad (2)$$

while the intensities /4/ are expressed in terms of the eigenvectors /3/ in the form

$$\psi(3/2,i) = \begin{vmatrix} C^{3/2}_{3/2,i} \\ C^{1/2}_{3/2,i} \\ C^{-1/2}_{3/2,i} \\ C^{-3/2}_{3/2,i} \end{vmatrix} = C^{m}_{3/2,i}\,;\ \psi(1/2,i) = \begin{vmatrix} C^{1/2}_{1/2,j} \\ C^{-1/2}_{1/2,j} \end{vmatrix} = C^{m'}_{1/2,j} \qquad (3)$$

$$P(\Theta,\varphi;\,3/2,i;\,1/2,j) = \left| \sum_{m,m'} C^{m'*}_{1/2,j} \cdot C^{m}_{3/2,i} \cdot M_{m'm} \right|^2 \qquad (4)$$

hence we get in the case of magnetic dipole transition

$$M_{m'm} = \langle 1/2\, m';\, L, m \,|\, 3/2, m \rangle \cdot X^{m}_{L}$$

The symbol $\langle\,/\,\rangle$ stands here for the Clebsch-Gordan coefficient, while X is the vector normal to the direction of the γ-radiation. It can be decomposed to the contributions

$$X'_\varphi = -\frac{1}{\sqrt{2}} e^{i\varphi} \qquad X^o_\varphi = 0$$

$$X'_\theta = \frac{i}{\sqrt{2}} \cos\theta e^{i\varphi} \qquad X^o_\theta = i\sin\theta$$

$$X^{-1}_\varphi = -\frac{1}{\sqrt{2}} e^{-i\varphi}$$

$$X^{-1}_\theta = -\frac{i}{\sqrt{2}} \cos\theta e^{-i\varphi}$$

In the case of polycrystalline samples, the crystal grains are randomly distributed, thus the intensities /4/ of the lines are evaluated by integration over θ and φ as

$$\bar{p}(3/2,i;1/2,j) = \int_0^\pi \int_0^{2\pi} p(\theta,\varphi;3/2,i;1/2,j) \cdot \sin\theta d\theta d\varphi \qquad /5/$$

The eigenvalues of the matrix /I/ can be easily evaluated and we get $TE_{1,2} = \pm 1/2\beta$

In the general case the characteristic equation for /II/ is given by the fairly complicated formula

$$X^4 + a\lambda^2 + b\lambda + c = 0 \qquad /6/$$

Where

$$a = -\left[18A^2(1 + 1/3\eta^2) + 5/2\alpha^2\right]$$

$$b = -12A\alpha^2\cos^2\theta - 3\eta A\alpha^2\sin^2\theta \cdot \cos^2\varphi + 6A\alpha^2\sin^2\theta$$

$$c = (3\eta^2A^2 - 3/4\alpha^2\sin^2\theta)^2 + 3\eta^2A^2(18A^2 + 3/2\alpha^2\cos^2\theta) +$$
$$+ \alpha^2\sin^2\theta(27A^2 + 9/8\alpha^2\cos^2\theta) + 18\eta A^2\alpha^2\sin^2\theta \cdot \cos 2\varphi +$$
$$+ 81A^4 + 9/16\alpha^4\cos^4\theta - 45/2 A^2\alpha^2$$

Practically, it is impossible to solve the equation /6/ for the general case. Let us see therefore the most important special cases namely, a/ $H \neq 0$, $\theta = \varphi = 0$

for which the eigenvalue equation /6/ has the form

$$\lambda^4 - 18A^2(1 + 1/3\eta^2)\lambda^2 + 81A^2(1 + 1/3\eta^2)^2 = 0 \qquad /7/$$

and $\lambda_{1,2} = 3A(1 + 1/3\eta^2)^{1/2}$

$\lambda_{3,4} = -3A(1 + 1/3\eta^2)^{1/2}$

b/ $A = 0$; $\theta = \varphi = 0$

then eq. /6/ becomes

$$\lambda^4 - 5/2\alpha^2\lambda^2 + 9/16\alpha^4 = 0 \qquad /8/$$

$\lambda_{1,2} = \pm 3/2\alpha$

$\lambda_{3,4} = \pm 1/2\alpha$

c/ $A \neq 0$; $\theta = \varphi = 0$; $\eta = 0$ and $A \ll \alpha$

with eq. /6/ reducing to

$$\lambda^4 - 5/2\alpha^2\lambda^2 - 12A\alpha^2\lambda + 9/16\alpha^4 = 0 \qquad /9/$$

We try to find the solution in the form $\lambda_i = \lambda_i^0 + \Delta\lambda_i$, where λ_i is the solution to eq. /8/ . Substituting λ_i into the expression /9/ and omitting the small, second order terms, we get $\Delta\lambda_i = \dfrac{12A\alpha^2}{4\lambda_i^{o2} - 5\alpha^2}$ from which the roots are obtained as

$$\lambda_1 = 3/2\,\alpha + 3A$$

$$\lambda_2 = 1/2\,\alpha - 3A$$

$$\lambda_3 = -1/2\,\alpha - 3A$$

$$\lambda_4 = -3/2\,\alpha + 3A$$

d/ $\varphi = 0$, $\eta = 0$, $A \neq 0$, $\theta \neq 0$ and $A \ll \alpha$

The form /6/ is then

$$\lambda^4 - 5/2\,\alpha^2 \lambda^2 + 6A\alpha^2 (1-3\cos^2\theta) + 9/16\,\alpha^4 = 0$$

Using again the substitution and ommission applied in the case c/, we find.

$$\Delta\lambda_i = \frac{6A\alpha^2(1-3\cos^2\theta)}{4\lambda_i^{o2} - 5\alpha^2}$$

and hence the roots can be defined as

$$\lambda_1 \doteq 3/2\,\alpha + 3/2\,A\,(1 - 3\cos^2/\theta/)$$

$$\lambda_2 = 1/2\,\alpha - 3/2\,A\,(1 - 3\cos^2/\theta/)$$

$$\lambda_3 = -1/2\,\alpha - 3/2\,A\,(1 - 3\cos^2/\theta/)$$

$$\lambda_4 = -3/2\,\alpha + 3/2\,A\,(1 - 3\cos^2/\theta/)$$

e/ If the magnetic field lies in the direction of one of the principal axes of the electric field gradient tensor, the eigenvalue equation can be written in the form

$$\lambda^4 - \left[18\,A^2(1 + 1/3\,\eta^2) + 5/2\,d^2\right]\lambda^2 - 12\,A\,d^2\lambda +$$
$$+ 81A^4(1 + 1/3\,\eta^2)^2 + 9/16\,d^4 + 9/2\,d^2A^2(\eta^2 - 5) \qquad /10/$$

The solutions to equations of the form $y^4 + py^2 + qy + r = 0$ can be written as

$$y_1 = 1/2\left(\sqrt{z_1} + \sqrt{z_2} + \sqrt{z_3}\right)$$

$$y_2 = 1/2\left(\sqrt{z_1} - \sqrt{z_2} - \sqrt{z_3}\right)$$

$$y_3 = 1/2\left(-\sqrt{z_1} + \sqrt{z_2} - \sqrt{z_3}\right)$$

$$y_4 = 1/2\left(-\sqrt{z_1} - \sqrt{z_2} + \sqrt{z_3}\right)$$

where z_i stands for the roots of the equation $z^3 + 2pz^2 + (p^2 - 4r)z - q^2 = 0$ which in the present case reads

$$z^3 - \left[5d^2 + 12A^2\eta^2 + 36A^2\right]z^2 + d^2\left[4d^2 + 12A^2\eta^2 + 180A^2\right]z - 144A^2d^4 = 0 \qquad /11/$$

On introducing $d^2 = a$, the rearrangement of /11/ with respect to the equal powers of the products of a and z, leads to

$$z(z^2-5az+4a^2)-12A^2\eta^2z(z-a)+3A^2(5az-z^2-4a^2)=(z-36A^2)$$
$$[z(z-a)-4a(z-a)]-12A^2\eta^2z(z-a)=0$$

that is, $z = a$ is a root of eq. /11/.

It is now easy to find the two other roots and the solution to /11/ can be written as

$$y_{1,2} = 1/2\alpha \pm \left[/3A+\alpha/^2+3A^2\eta^2 \right]^{1/2}$$

$$y_{3,4} = -1/2\alpha \pm \left[/3A-\alpha/^2 + 3A^2\eta^2 \right]^{1/2}$$

With this we have actually given all the possible analytical solutions to eq. /6/. If none of the assumptions made in a/, b/, c/, d/, or e/ is valid, only numerical methods can help. The numerical solutions can be then utilized for the evaluation of the positions of the hyperfine lines in the Mössbauer spectra by making use of the formula /2/.

§ 2. The transition probabilities can be given /3/ in tabulated form as follows.

for $H = 0$, $A \neq 0$

Transition	I/Θ/	I/polycrystal/
$\pm 3/2 \rightarrow \pm 1/2$	$3/2(1+\cos^2\theta)$	1
$\pm 1/2 \rightarrow \pm 1/2$	$1+3/2\sin^2\theta$	1

where θ is the angle between the symmetry axis and the direction of the γ-ray.

for $H \gg A$

Transition	I/θ/	I/polycrystal/
$3/2 \rightarrow 1/2$ $-3/2 \rightarrow -1/2$	$9/4(1+\cos^2\theta)$	3
$1/2 \rightarrow 1/2$ $-1/2 \rightarrow -1/2$	$3\sin^2\theta$	2
$-1/2 \rightarrow 1/2$ $1/2 \rightarrow -1/2$	$3/4(1+\cos^2\theta)$	1

where θ is the angle between the magnetic field and the direction of the γ-ray.

§ 3. In the computer programs the method of least squares developed by Silin /4/ was applied. The ALGOL and FORTRAN versions of the method are presented in the Annexes I. and II. The advantages of the method are its clarity and the fact that it can be applied to different problems upon resriting only a single, namely the "sqz" procedure.

The following notations are used

$$S = 1/2 \sum_i \left(\frac{y_i - f_i}{sig_i} \right)^2$$

where y_i are the measured points

f_i are the calculated values

sig_i are the errors of the experimental points.

$$Z_{ik} = \frac{\partial^2 f}{\partial \alpha_i \partial \alpha_k}$$

n = the number of parameters

n_1 = maximum number of steps within an iteration

n_2 = minimum number of iterations without

n_3 = maximum number of iterations after the program stops irrespective of the accuracy.

eps = accuracy / if kappa $\leq$ abs /eps/, the program stops

$a[i]$ = the value of the parameters

$l[i]$ = during an iteration the correction of the parameters cannot be greater than $l[i]$. The starting values are specified by the programmer.

sigma/i/ = sqrt/Z_{ii}/ = the statistical errors of the parameters

$r[i]$ = correlation factors

da/i/ = the corrections of the parameters = - lambda /$z^{-1}g$/$_i$

$$\text{kappa} = \max_i \left| \frac{(z^{-1}g)_i}{sigma[i]} \right|$$

$$\text{lambda} = \frac{1}{\max_i \left(1, \left| \frac{(z^{-1}g)_i}{l[i]} \right| \right)}$$

procedures :

fumilim - dispatcher part of the program

monitor - printout of partial results

mconv - inversion of the symmetrical matrix

mvmult - calculation of the $da = \left(z^{-1} g \right)_i$ product

fix - fixing of the parameter values for which

$1/i/ = 0$

sgz - calculation of the values of s, g, and z

§ 4. In this paragraph the /sgz/ procedures for the different practical cases are described.

a/ The spectrum is assumed to be the superposition of N Lorentzian curves, and can be thus described by the function

$$f_j = \left(A_o + Bx_j\right)\left[1 - \sum_{k=1}^{N} \frac{a_k}{(x_j - \delta_k)^2 + \gamma_k^2/4}\right] \qquad /12/$$

the sequence of the parameters is

A_o - is the ground-line

a_k - N magnitudes

δ_k - N line-positions

γ_k - N line-widths

B - the slope of the ground line

b/ The assumption a/ imposes often difficult problems /e.g. the presence of many poorly resolved peaks to the program. In this case use can be made of the expression in § 1.c/ which describes the relative positions of the six lines in the so called 6-line Zeeman pattern. If the spectrum is produced by the superposition of W Zeeman patterns, it can be described by the function

$$f(x_j) = (A_0 + Bx_j)\left[1 - \sum_{i=1}^{6}\sum_{k=1}^{W} \frac{a_k \cdot k_i}{(x_j - \delta_{ik})^2 + \gamma^2/4}\right] \qquad /13/$$

where $\delta_{ik} = \delta_{ik}(\varepsilon_k, H_k, \delta_k)$

the sequence of the parameters is

γ - is the line-width,

A_0 - the ground line,

a_k - W magnitudes,

H_k - W magnetic fields /in $\frac{\text{kiloGauss}}{\text{mm/sec}}$/

δ_k - W line-positions /in channels/,

ε_k - W quadrupole splittings /in channels/,

B - the slope of the groundline,

k_i - the relative magnitude of the Zeeman pattern lines,

a, b, c - the coefficients at Hi for the Zeeman pattern lines /e.g. for Fe^{57}: 0.01615, 0.00925, 0.00258/

/Annexes III. and IV./

c/ Finally, if the numerical solution of the eigenvalue problem is the only possible method, the program described in Annexe V can be used. This program is suitable for the least square fit of two 6-line Zeeman patterns and two 8-line patterns to the function

$$f(x_j) = (A_0 + Bx_j)\Big[1 - \sum_{i=3}^{4} a_i\Big(\frac{K_1}{(x_j + aH_i - \delta_i + \varepsilon_i)^2 + \gamma_i^2/4} + \frac{K_2}{(x_j + bH_i - \delta_i - \varepsilon_i)^2 + \gamma_i^2/4} + \frac{K_3}{(x_j + cH_i - \delta_i - \varepsilon_i)^2 + \gamma_i^2/4} + \frac{K_3}{(x_j - cH_i - \delta_i - \varepsilon_i)^2 + \gamma_i^2/4} + \frac{K_2}{(x_j - bH_i - \delta_i - \varepsilon_i)^2 + \gamma_i^2/4} + \frac{K_1}{(x_j - aH_i - \delta_i + \varepsilon_i)^2 + \gamma_i^2/4}\Big) - \sum_{k=1}^{2} a_k\Big(\frac{P_{11k}}{(x_j - \delta_k - T_{11k})^2 + \gamma_k^2/4} + \frac{P_{12k}}{(x_j - \delta_k - T_{12k})^2 + \gamma_k^2/4} + \frac{P_{13k}}{(x_j - \delta_k - T_{13k})^2 + \gamma_k^2/4} + \frac{P_{14k}}{(x_j - \delta_k - T_{14k})^2 + \gamma_k^2/4} + \frac{P_{21k}}{(x_j - \delta_k - T_{21k})^2 + \gamma_k^2/4} + \frac{P_{22k}}{(x_j - \delta_k - T_{22k})^2 + \gamma_k^2/4} + \frac{P_{23k}}{(x_j - \delta_k - T_{23k})^2 + \gamma_k^2/4} + \frac{P_{24k}}{(x_j - \delta_k - T_{24k})^2 + \gamma_k^2/4}\Big)\Big]$$

the sequence of the parameters is

A_0 - the ground-line

$\left.\begin{matrix}\gamma_k \\ \gamma_i\end{matrix}\right\}$ - four line-widths /in channels/

$\left.\begin{matrix}a_k \\ a_i\end{matrix}\right\}$ - four magnitudes

H_k - two magnetic fields /in kiloGauss/

H_i - two magnetic fields /in $\frac{kG \cdot channel}{mm/sec}$/

$\left.\begin{matrix}\delta_k \\ \delta_i\end{matrix}\right\}$ - four isomer shifts /in channels/

$\mathcal{E}_k$ - two quadrupole splitting $\mathcal{E} = A$ /in channels/

$\mathcal{E}_i$ - two quadrupol splitting $\mathcal{E} = \frac{1}{4} e^2 Q V_{zz}$ /in channels/

$\left.\begin{matrix}Q_k \\ \varphi_k\end{matrix}\right\}$ - the polase wordinates /in radians/

η_k - two assymetry parameters

B - the slope of the ground-line

for the i-th components one must give the coefficients K and a,b,c /see § 4.b/

Annexe V.

§ 5. An ever mor frequent problem is the evaluation of the contribution from the crystal field to the electric field gradient tensor for given lattice parameters and charge distribution.

The contribution to the tensor V_{ik} can be defined as

$$V_{xx} = \sum \frac{1}{r^3}\left(\frac{3x^2}{r^2} - 1\right); \quad V_{xy} = V_{yx} = \sum \frac{3xy}{r^5};$$

$$V_{yy} = \sum \frac{1}{r^3}\left(\frac{3y^2}{r^2} - 1\right); \quad V_{xz} = V_{zx} = \sum \frac{3xz}{r^5};$$

$$V_{zz} = \sum \frac{1}{r^3}\left(\frac{3z^2}{r^2} - 1\right); \quad V_{yz} = V_{zy} = \sum \frac{3yz}{r^5};$$

The program, which works for orthogonal coordinate systems, only, calculates the lattice sum of the tensor V^i_{ik} up to the desired coordination sphere. Then it diagonalizes the sum and evaluates in addition to the eigenvalues, the eigenvectors, too. Finally it yields the values of the quadrupole splitting $\mathcal{E} = \frac{1}{2} e^2 V_{zz} \cdot Q \cdot (1 + 1/3\,\eta^2)^{1/2}$ and the asymmetry parameter $\eta = \frac{V_{xx} - V_{yy}}{V_{zz}}$

Annexes VI. - VII.

the sequence of parameters for the Annexe VI. is

N - is the number of atom in the unit cello,

NA - the number of steps with the change of the lattice parameters,

NW - the number of the atom for which the EFG tensor will be computed,

A,B,C - the lattice parameters,

V - the nuclear quadrupole moment,

L_1, L_2, L_3 - the value of the steps of the lattice parameter change,

M - the number of neighboring unit cells around the central one in the direction x, y, z

x_i, y_i, z_i - the coordinats of the atoms in the unit cell

Si - the charges of the atoms in the unit cell

the sequence of the parameters for the Annexe VII.

nn - the number of the repetion of the evaluation

n - N

xa- NA

xb- the number can be used to identify the users program

w - NW

a,b,c, - A,B,C .etc.

Literature

[1] A.Abragam L'effet Mössbauer et ses applications a l'étude des champs internes /Gordon and Breach, Science Publ.,Inc., New York 1964/

[2] W.Kündig, Nuclear Instruments and Methods 48 /1967/ 219-228

[3] Wertheim G.K. Mössbauer Effect Principles and Applications /Academic Press New York and London/

[4] И.Н. Силин., Стандартная программа для решения задач методом наименьших квадратов. Препринт № II-3362. Дубна. ОИЯИ

Figure Captions

Fig 1. The polar coordinates of the internal magnetic field relative to the principal axes of the electric field gradient tensor

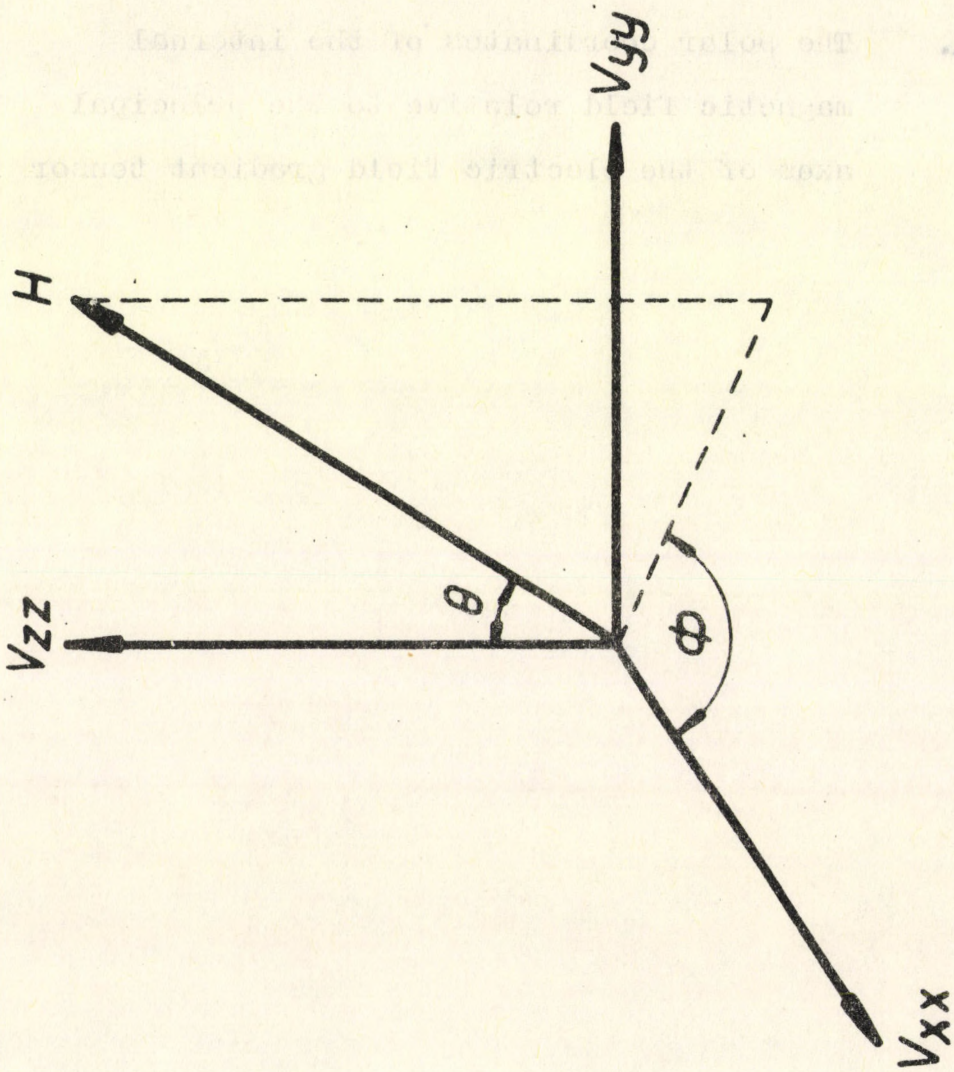

Fig. 1

Annexes I.

```
'procedure' form(z,g,df,y,sig,n);
      'value' y,sig,n;'real' y,sig;
      'integer' n;'real' 'array' z,g,df;
      'begin' 'integer' i,k,l;
          l:=1;y:=y/sig;
      'for' i:=1 'step' 1 'until' n 'do'
       'begin' g[i]:=g[i]+df[i]*y/sig;
      'for' k:=1 'step' 1 'until' i 'do'
      'begin' z[l]:=z[l]+df[i]/sig*
             (df[k]/sig);l:=l+1;
    'end' k;
'end' i;
'end' form;


'procedure' fiX(z,l,c,n);'value' c,n;
       'real' 'array' z,l;'real' c;'integer' n;
     'begin' 'integer' i,k;
      'for' i:=1 'step' 1 'until' n 'do'
      'if' l[i]=0 'then'
     'for' k:=1 'step' 1 'until' n 'do'
      z['if' k'le'i 'then' i*(i-1)/2+k 'else' k*(k-1)/2+i]:=
      'if' i#k 'then' 0 'else' c;
'end' fiX;
```

```
'procedure' mconv(z,n):
      'value' n;'integer' n;'real' 'array' z;
     'begin' 'integer' i,k,l;'real' c,d;
     diagonalization:
     'for' i:=1 'step' 1 'until' n-1 'do'
      'for' k:=n 'step' -1 'until' i+1 'do'
     'begin'
     c:=z[k*(k-1)/2+i]/z[i*(i+1)/2];
      'for' l:=k 'step' -1 'until' 1 'do'
      z[k*(k-1)/2+l]:=z[k*(k-1)/2+l]-
      z['if' l>i 'then' l*(l-1)/2+i
                'else' i*(i-1)/2+l]*c;
      z[k*(k-1)/2+i]:=-c
   'end' i,k;
     multiplication:
     'for' i:=1 'step' 1 'until' n 'do'
     'for' k:=i 'step' 1 'until' n 'do'
  'begin' d:=0;
            'for' l:=k 'step' 1 'until' n 'do'
  'begin'
            c:=1/z[l*(l+1)/2];
            'if' l#i 'then'
            c:=c*z[l*(l-1)/2+i];
            'if' l#k 'then'
            c:=c*z[l*(l-1)/2+k];
            d:=d+c;
   'end' l;
        z[k*(k-1)/2+i]:=d;
    'end' i,k
'end' mconv;
```

```
'procedure' mvmult(z,g,da,n);
      'value' n;'integer' n;'real' 'array' z,g,da;
    'begin' 'integer' i,l;
      'for' i:=1 'step' 1 'until' n 'do'
      'begin' da[i]:=0;
      'for' l:=1'step' 1 'until' n 'do'
       da[i]:=da[i]+g[l]*z['if' i'ge'l 'then'
       i*(i-1)/2+l 'else' l*(l-1)/2+i];
   'end' i,l;
'end' mvmult;


'procedure' fumilim(s,g,z,n,n1,n2,n3,eps,a,l,sigma,r,da,kapPa,lambda,
            sgz,monitor);
       'value' n,n ,n2,n3,eps;'integer' n,n1,n2,n3;
       'real' s,eps,kappa,lambda;
       'real' 'array' g,z,a,l,sigma,r,da;
       'procedure' sgz,monitor;
     'begin' 'real' t,maX,gt,olds,t1;
      'integer' i,nn1,nn2,nn3;
        nn2:=0;nn3:=0;
        sgz(s,g,z,a,n);t1:=1;
        'goto' shift;
      control:nn1:=1;t1:=1;
      repeat:sgz(s,g,z,a,n);
             'if' nn1>n1 'then' 'goto' shift;
             t:=2*(s-olds-gt);
             'if' 0.59*t+gt'le'0 'then' 'goto' shift;
             t:=-1.5*gt/t;'if' t<0.25 'then' t:=0.25;
             gt:=gt*t;t1:=t1*t;nn2:=0;
       'for'i:=1'step'1'until'n'do'
         'begin' a[i]:=a[i]-da[i];l[i]:=l[i]*t;
         da[i]:=da[i]*t;a[i]:=a[i]+da[i];
         'end' i;nn1:=nn1+1;
         'goto' repeat;
   shift:fiX(z,l,1,n);
  'for' i:=1 'step' 1 'until' n 'do'
         r[i]:=z[i*(i+1)/2];
         mconv(z,n);
```

```
          'for' i:=1 'step' 1 'until' n 'do'
          r[i]:=r[i]*z[i*(i+1)/2];
          fiX(z,1,0,n);
          'if' eps>0 'then'
          'for' i:=1 'step' 1 'until' n 'do'
          sigma[i]:=sqrt(z[i*(i+1)/2]);
          mvmult(z,g,da,n);
          maX:=1;kappa:=0;
          'for' i:=1 'step' 1 'until' n 'do' 'if' l[i]#0
          'then' 'begin'
          'if' abs(da[i]/l[i])>maX 'then'
          maX:=abs(da[i]/l[i]);
          'if' abs(da[i]/sigma[i])>kappa 'then'
          kappa:=abs(da[i]/sigma[i]);
          'end' i;lambda:=1/maX;gt:=0;
         'for' i:=1 'step' 1 'until' n 'do' 'if' l[i]#0
          'then' 'begin' 'if' nn2'ge'n2 'then' 'begin'
          'if' abs(da[i]/l[i])>0.25*maX
          'then' l[i]:=4*l[i];'end';
          da[i]:=-da[i]*lambda;
           gt:=gt+da[i]*g[i]
         'end' i;
'if' test(2) 'then' 'go to' L1;
   'if' nn3=0'or' nn3=n3 'or' kappa'le' 0.2 'then'
L1:monitor(s,g,z,n,nn3,eps,a,l,sigma,r,da,gt,kappa,lambda,t1);
         'if' kappa'le'abs(eps) 'or'nn3'ge'n3 'or' gt 'ge' 0
   'then' 'goto' eXit;
         'for' i:=1 'step' 1 'until' n 'do'
         a[i]:=a[i]+da[i];
   olds:=s;nn2:=nn2+1;nn3:=nn3+1;
    'goto' control;
eXit:'end' fumilim;
```

Annexes II.

```
        SUBROUTINE FORM(Z,G,F,Y,SIG,N,N9)
        DIMENSION DF(N),G(N),Z(N9)
        L=1
        Y=Y/SIG
        DO 20 I=1,N
        G(I)=G(I)+DF(I)*Y/SIG
        DO 299 K=1,I
        Z(L)=Z(L)+DF(I)/SIG*(DF(K)/SIG)
        L=L+1
299     CONTINUE
20      CONTINUE
        RETURN
        END
```

```
        SUBROUTINE FIX(Z,L,C,N,N9)
        INTEGER C
        REAL L
        DIMENSION Z(N9),L(N)
        DO21 I=1,N
        IF(L(I)) 23,22,23
22      DO 296 K=1,N
        IF(K-I) 24,24,25
24      IF(I-K) 26,27,26
26      Z(I*(I-1)/2+K)=0
        GO TO 28
27      Z(I*(I-1)/2+K)=C
        GO TO 28
25      IF(I-K) 29,297,29
29      Z(K*(K-1)/2+I)=0
        GO TO 28
297     Z(K*(K-1)/2+I)=C
28      CONTINUE
296     CONTINUE
23      CONTINUE
21      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE MCONV(Z,N,N9)
      DIMENSION Z(N9)
      DO 30 I=1,N-1
      DO 30 K=I+1,N
      M=N-K+1+I
      C=Z(M*(M-1)/2+I)/Z(I*(I+1)/2)
      DO 33 L=1,M
      J=M-L+1
      IF (J-I) 32,32,31
31    Z(M*(M-1)/2+J)=Z(M*(M-1)/2+J)-Z(J*(J-1)/2+I)*C
      GO TO 33
32    Z(M*(M-1)/2+J)=Z(M*(M-1)/2+J)-Z(I*(I-1)/2+J)*C
33    CONTINUE
      Z(M*(M-1)/2+I)=-C
30    CONTINUE
      DO 34 I=1,N
      DO 314 K=I,N
      D=0
      DO 35 L=K,N
      C=1/Z(L*(L+1)/2)
      IF(L-I) 36,37,36
36    C=C*Z(L*(L-1)/2+I)
37    IF (L-K) 38,39,38
38    C=C*Z(L*(L-1)/2+K)
39    D=D+C
35    CONTINUE
      Z(K*(K-1)/2+I)=D
314   CONTINUE
34    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MVMULT(Z,G,DA,N,N9)
      DIMENSION Z(N9),G(N),DA(N)
      DO 40 I=1,N
      DA(I)=0
      DO 40 L=1,N
      IF(I-L) 41,42,42
42    DA(I)=DA(I)+G(L)*Z(I*(I-1)/2+L)
      GO TO 40
41    DA(I)=DA(I)+G(L)*Z(L*(L-1)/2+I)
40    CONTINUE
      RETURN
      END


      SUBROUTINE FUMILIM(G,Z,N,N1,N2,N3,EPS,A,L,SIGMA,R,DA,KAPPA,
     CLAMBDA,BETA,Y,B,TETA,N9,W,SZ,CIS)
      INTEGER B,ETA,TETA,W,CIS,Y
      REAL KAPPA,LAMBDA,T,MAX,T1,L
      DIMENSION G(N),A(N),L(N),SIGMA(N),R(N),DA(N),Z(N9),
     CBETA(CIS),Y(CIS),SZ(9)
      S=0
      NN2=0
      NN3=0
      CALL SGZ(S,G,Z,A,N,Y,BETA,B,L,N9,W,SZ,CIS)
      T1=1
      GO TO 50
51    NN1=1
      T1=1
```

```
52      CALL SGZ(S,G,Z,A,N,Y,BETA,B,L,N9,W,SZ,CIS)
        IF (NN1-N1) 53,53,54
54      GO TO 50
53      T=(S-OLDS-GT)*2.0
        FI=0.59*T+GT
        IF (FI) 50,50,56
56      T=-1.5*GT/T
        IF (T-0.25) 55,57,57
55      T=0.25
57      GT=GT*T
        T1=T1*T
        NN2=0
        DO 58 I=1,N
        A(I)=A(I)-DA(I)
        L(I)=L(I)*T
        DA(I)=DA(I)*T
        A(I)=A(I)+DA(I)
58      CONTINUE
        NN1=NN1+1
        GO TO 52
50      CALL FIX(Z,L,1,N,N9)
        DO 59 I=1,N
59      R(I)=Z(I*(I+1)/2)
        CALL MCONV(Z,N,N9)
        DO 60 I=1,N
60      R(I)=R(I)*Z(I*(I+1)/2)
        CALL FIX(Z,L,0,N,N9)
        IF (EPS) 61,61,62
62      DO 63 I=1,N
63      SIGMA(I)=SQRT(Z(I*(I+1)/2))
61      CALL MVMULT(Z,G,DA,N,N9)
        MAX=1.0
        KAPPA=0
        DO 645 I=1,N
        IF(L(I)) 65,64,65
65      FI=ABS(DA(I)/L(I))
        IF (FI-MAX) 67,67,66
66      MAX=FI
67      FI1=ABS(DA(I)/SIGMA(I))
        IF (FI1-KAPPA) 64,64,68
```

```
68     KAPPA=FI1
64     CONTINUE
645    CONTINUE
       LAMBDA=1/MAX
       GT=0
       DO 705 I= ,N
       IF(L(I)) 71,70,71
71     IF(NN2-N2)72,84,84
84     FI=ABS(DA(I)/L(I))
       IF (FI-0.25*MAX) 72,72,73
73     L(I)=L(I)*4.0
72     CONTINUE
       DA(I)=-DA(I)*LAMBDA
       GT=GT+DA(I)*G(I)
70     CONTINUE
705    CONTINUE
       IF (NN3) 74,75,74
74     IF (NN3-N3) 76,75,76
76     IF (KAPPA-0.2) 75,75,77
75     CALL MONITOR(S,N,NN3,A,SIGMA,R,KAPPA,LAMBDA,TETA)
77     IF (KAPPA-ABS(EPS)) 78,78,79
79     IF (NN3-N3) 80,78,78
80     IF (GT) 81,78,78
81     DO 82 I=1,N
82     A(I)=A(I)+DA(I)
       OLDS=S
       NN2=NN2+1
       NN3=NN3+1
       GO TO 51
78     CONTINUE
       RETURN
       END
```

Annexes III.

```
'procedure' Sgz(S,g,z,a,n); 'value' n; 'real' S; 'integer' n;
      'array' g,z,a;
'begin' 'integer'i,t;
      'array'df,ddf[1:n],d[1:128];
      'real'f,e,c,ma;
         S:=0;
        'for' i:=1 'step' 1 'until' n 'do'
         g[i]:=0; t:=n*(n+1)/2;
        'for' i:=1 'step' 1 'until' t 'do'
         z[i]:=0;
    f:=w+1;
    e:=2*w+1;
        'for' t:=6'step' 1 'until' 123 'do'
'begin'beta[t]:=0;
     ma:=a[n]*x[t]+a[1];
    'for' i:=1 'step' 1 'until'w 'do'
'begin'
  c:=(x[t]-a[f+i])↑2+a[e+i]↑2/4;
  beta[t]:=beta[t]+a[i+1]/c;
  df[i+1]:=-ma/c;
  df[f+i]:=-2*ma*a[i+1]*(x[t]-a[f+i])/c↑2;
  df[e+i]:=ma*a[i+1]*a[e+i]/(2*c↑2);
'end';
  df[1]:=1-beta[t];
  df[n]:=x[t]*(1-beta[t]);
      'for'i:=1'step'1'until'n'do'
           ddf[i]:=df[i];
     'if' eta=7  'then'
'begin'  ddf[2*w+2]:=0;
    'for' i:=1'step' 1'until' w'do'
        ddf[2*w+2]:=ddf[2*w+2]+df[2*w+i+1];
    'for' i:=1'step' 1'until' w-1 'do'
'begin'   ddf[2*w+2+i]:=0;
```

```
        l[2*w+2+i]:=0;
        a[2*w+2+i]:=a[2*w+2];
'end';
'end';
     beta[t]:=0;
        'for' i:=1 'step' 1 'until'w'do'
      beta[t]:=beta[t]+a[i+1]/((x[t]-a[w+i+1])↑2
             +a[2*w+i+1]↑2/4);
  beta[t]: ma*(1-beta[t]);
        'if' y[t]=0 'then' y[t]:=eta[t];
        d[t]:=sqrt(y[t]);
form(z,g, d df,beta[t]-y[t],d[t],n);
S:=S+((beta[t]-y[t])/d[t])↑2
   'end'
 'end';
```

```
      SUBROUTINE SGZ(S,G,Z,A,N,Y,ETA,BETA,B,L,N9)
      INTEGER T,W,B,ETA,F,E
      REAL MA,L
      DIMENSION DF(30),DDF(30),BETA(128),G(N),
     CA(N),L(N),Z(N9),Y(128)
      S=0
      DO 85 I=1,N
85    G(I)=0
      T=N*(N+1)/2
      DO 86 I=1,T
86    Z(I)=0
      W=(N-2)/3
      F=W+1
      E=2*W+1
      DO 87 T=6,123
      BETA(T)=0
      MA=A(N)*T+A(1)
      DO 88 I=1,W
      C=(T-A(F+I))**2+A(E+I)**2/4
      BETA(T)=BETA(T)+A(I+1)/C
      DF(I+1)=-MA/C
      DF(F+I)=-MA*2.0*A(I+1)*(T-A(F+I))/C**2
      DF(E+I)=MA*A(I+1)*A(E+I)/(2*C**2)
88    CONTINUE
      DF(1)=1.0-BETA(T)
      DF(N)=T*(1-BETA(T))
      DO 89 I=1,N
89    DDF(I)=DF(I)
      IF(ETA-7) 855,91,855
91    DDF(2*W+2)=0
      DO 92 I=1,W
92    DDF(2*W+2)=DDF(2*W+2)+DF(2*W+I+
      DO 90 I=1,W-1
      DDF(2*W+2+I)=0
      L(2*W+2+I)=0
      A(2*W+2+I)=A(2*W+2)
90    CONTINUE
```

```
855   BETA(T)=0
      DO 93 I=1,W
93    BETA(T)=BETA(T)+A(I+1)/((T-A(F+I))**2+A(E+I)**2/4)
      BETA(T)=MA*(1-BETA(T))
      DT=SQRT(Y(T))
      DTT=BETA(T)-Y(T)
      CALL FORM(Z,G,DDF,DTT,DT,N,N9)
      S=S+(BETA(T)-Y(T))**2/Y(T)
87    CONTINUE
      RETURN
      END
```

Annexes IV.

```
'procedure' Sgz(S,g,z,a,n); 'value' n; 'real' S;'integer' n;
          'array' g,z,a;
'begin''integer'jk,nn k,j,i,t; .
      'array' ddf,df[1:n];
      'integer' aa,bb cc;
           'real' kk,tt,mm,yy,xx,zz,ky,ma,ki,ti,mi,yi,xi,zi;
         S:=0; kv:=a[1]↑2/4;
        'for'i:=1'step'1'until'n'do'
g[i]:=0;
t:=n*(n+1)/2;
         'for' i:=1 'step' 1 'until' t 'do'
             z[i]:=0;
          'for' t:=6'step' 1 'until' 123'do'
'begin' beta[t]:=0;
         ma:=a[2]+a[n]*t;
     df[1]:=0;
      'for'i:=1'step'1'until'w'do'
'begin' aa:=2+w+i;
        bb:=2+2*w+i;
        cc:=2+3*w+i;
kk:=t+kfi*a[aa]-a[bb]+a[cc];
tt:=t+kfe*a[aa]-a[bb]-a[cc];
mm:=t+kff*a[aa]-a[bb]-a[cc];
yy:=t-kff*a[aa]-a[bb]-a[cc];
xx:=t-kfe*a[aa]-a[bb]-a[cc];
zz:=t-kfi*a[aa]-a[bb]+a[cc];
ki:=sza/(kk↑2+ky);
ti:=szb/(tt↑2+ky);
mi:=szc/(mm↑2+ky);
yi:=szd/(yy↑2+ky);
xi:=sze/(xx↑2+ky);
zi:=szf/(zz↑2+ky);
df[i+2]:=ki+ti+mi+yi+xi+zi;
beta[t]:=beta[t]+a[i+2]*df[i+2];
df[i+2]:=-ma*df[i+2];
ki:=ki/(kk↑2+ky);
ti:=ti/(tt↑2+ky);
mi:=mi/(mm↑2+ky);
yi:=yi/(yy↑2+ky);
xi:=xi/(xx↑2+ky);
zi:=zi/(zz↑2+ky);
   df[1]:=df[1]+a[i+2]*(ki+ti+mi+yi+xi+zi);
kk:=ki*kk;
tt:=ti*tt;
mm:=mi*mm;
yy:=yi*yy;
xx:=xi*xx;
zz:=zi*zz;
ki:=2*ma*a[i+2];
   df[bb]:=-ki*(kk+tt+mm+yy+xx+zz);
   df[aa]:=-ki*(-kk*kfi-tt*kfe-mm*kff
                +yy*kff+xx*kfe+zz*kfi);
   df[cc]:=ki*(kk-tt-mm-yy-xx+zz);
```

```
'end';
    df[2]:=1-beta[t];
    df[n]:=t*df[2];
    df[1]:=ma*a[1]/2*df[1];
      'for'i:=1'step'1'until'n'do'
      ddf[i]:=df[i];
    'if' eta[1]=111
        'then' 'goto' cim;
   'for' k:=1 'step' w 'until' 4*w 'do'
'begin'  'if' eta[k]=5
         'then'
'begin'      ddf[k+2]:=0;
        'for'nn:=0'step'1'until'w-1'do'
              ddf[k+2]:=ddf[k+2]+df[k+2+nn];
        'for'nn:=1'step'1'until'w-1'do'
'begin'  ddf[k+2+nn]:=0;
         l[k+2+nn]:=0;
         a[k+2+nn]:=a[k+2];
'end';
   'goto' metka;
'end';
    'if' w=4 'then'
'begin'
            'for'i:=1'step'1'until'2'do'
                'for'j:=2'step'1'until'3'do'
                    'for'nn:=3'step'1'until'4'do'
                        'for'jk:=4'step'1'until'5'do'
'begin'
         'if'(eta[k+i-1]=eta[k+j-1]=eta[k+nn-1]=eta[k+jk-1])
         'and' i<j  'and' j<nn  'and' nn<jk
      'then'
'begin' ddf[k+1+i]:=df[k+1+j]+df[k+1+i]+df[k+1+jk]+df[k+1+nn];
        ddf[k+1+j]:=0;
        ddf[k+1+nn]:=0;
        ddf[k+1+jk]:=0;


        l[k+1+j]:=0;
        l[k+1+nn]:=0;
        l[k+1+jk]:=0;
        a[k+1+j]:=a[k+1+i];
        a[k+1+nn]:=a[k+1+i];
        a[k+1+jk]:=a[k+1+i];
   'goto' metka;
'end';
'end';
'end';
    'if' w=3 'then'
'begin'
         'for'i:=1'step'1'until'3'do'
              'for'j:=2'step'1'until'4'do'
                   'for'nn:=3'step'1'until'5'do'
'begin'   'if'(eta[k-1+i]=eta[k-1+j]=eta[k-1+nn])
          'and'  i<j   'and'  j<nn
           'then'
```

```
'begin'
        ddf[k+1+i]:=df[k+1+j]+df[k+1+i]+df[k+1+nn];
        ddf[k+1+j]:=0;
        ddf[k+1+nn]:=0;
        l[k+1+j]:=0;
        l[k+1+nn]:=0;
        a[k+1+j]:=a[k+1+i];
         a[k+1+nn]:=a[k+1+i];
   'goto' metka;
'end';
'end';
'end';
    'if' w=2 'then'
'begin'
     'for'i:=0'step'1'until'3'do'
       'for'j:=2'step'1'until'5'do'
'begin'
'if'eta[k+i]=eta[k+j-1]
'and'(k+j-1)>(k+i)
'then'
'begin'ddf[i+k+2]:=df[k+j+1]+df[i+k+2];
       ddf[k+1+j]:=0;
       a[k+1+j]:=a[i+k+2];
       l[k+1+j]:=0;
'end'
'end';
'end';
    metka:'end';
cim: beta[t]:=0;
   'for' i:=1'step'1'until' w 'do'
'begin'  aa:=2+w+i;
         bb:=2+2*w+i;
         cc:=2+3*w+i;
  beta[t]:=beta[t]+a[i+2]*(
   sza/((t+kfi*a[aa]-a[bb]+a[cc])↑2+ky)+
   szb/((t+kfe*a[aa]-a[bb]-a[cc])↑2+ky)+
   szc/((t+kff*a[aa]-a[bb]-a[cc])↑2+ky)+
   szd/((t-kff*a[aa]-a[bb]-a[cc])↑2+ky)+
   sze/((t-kfe*a[aa]-a[bb]-a[cc])↑2+ky)+
   szf/((t-kfi*a[aa]-a[bb]+a[cc])↑2+ky));
'end';
   beta[t]:=ma*(1-beta[t]);
           'if' y[t]=0'then' y[t]:=beta[t];
   ma:=sqrt(y[t]);
   kk:=beta[t]-y[t];
   form(z,g ddf,kk,ma,n);
   S:=S+(kk/ma)↑2;
'end'
'end'+
```

```
      SUBROUTINE SGZ(S,G,Z,A,N,Y,BETA,B,L,N9,W,SZ,CIS)
      INTEGER T,W,B,AA,BB,CC,CIS,Y
      REAL S,MA,L,KK,TT,MM,YY,XX,ZZ,KY,KI,TI,MI,YI,XI,ZI
      DIMENSION G(N),Z(N9),A(N),Y(CIS),BETA(CIS),L(N),SZ(9),DF(23)
      S=0.0
      KY=A( )**2/4
      DO 854 I=1,N
854   G(I)=0
      DO 855 I=1,N9
855   Z(I)=0
      II=CIS-5
      DO 856 T=6,II
      BETA(T)=0
      MA=A(2)+A(N)*T
      DF( )=0
      DO 857 I=1,W
      AA=2+W+I
      BB=2+2*W+I
      CC=2+3*W+I
      KK=T+SZ(1)*A(AA)-A(BB)+A(CC)
      TT=T+SZ(2)*A(AA)-A(BB)-A(CC)
      MM=T+SZ(3)*A(AA)-A(BB)-A(CC)
      YY=T-SZ(3)*A(AA)-A(BB)-A(CC)
      XX=T-SZ(2)*A(AA)-A(BB)-A(CC)
      ZZ=T-SZ(1)*A(AA)-A(BB)+A(CC)
      KI=SZ(4)/(KK**2+KY)
      TI=SZ(5)/(TT**2+KY)
      MI=SZ(6)/(MM**2+KY)
      YI=SZ(7)/(YY**2+KY)
      XI=SZ(8)/(XX**2+KY)
      ZI=SZ(9)/(ZZ**2+KY)
```

```
      DF(I+2)=KI+TI+MI+YI+XI+ZI
      BETA(T)=BETA(T)+A(I+2)*DF(I+2)
      DF(I+2)=-MA*DF(I+2)
      KI=KI/(KK**2+KY)
      TI=TI/(TT**2+KY)
      MI=MI/(MM**2+KY)
      YI=YI/(YY**2+KY)
      XI=XI/(XX**2+KY)
      ZI=ZI/(ZZ**2+KY)
      DF(1)=DF(1)+A(I+2)*(KI+TI+MI+YI+XI+ZI)
      KK=KI*KK
      TT=TI*TT
      MM=MI*MM
      YY=YI*YY
      XX=XI*XX
      ZZ=ZI*ZZ
      KI=2.0*MA*A(I+2)
      DF(BB)=-KI*(KK+TT+MM+YY+XX+ZZ)
      DF(AA)=-KI*(-KK*SZ(1)-TT*SZ(2)-MM*SZ(3)+YY*SZ(3)+XX*SZ(2)
     C+ZZ*SZ(1))
857   DF(CC)=KI*(KK-TT-MM-YY-XX+ZZ)
      DF(2)=1-BETA(T)
      DF(N)=T*DF(2)
      DF( )=MA*A(1)/2*DF( )
      BETA(T)=MA*(1-BETA(T))
      IF(Y(T)) 861,860,861
860   Y(T)=BETA(T)
861   MA=SQRT(Y(T))
      KK=BETA(T)-Y(T)
      CALL FORM(Z,G,DF,KK,MA,N,N9)
      S=S+(BETA(T)-Y(T))**2/Y(T)
856   CONTINUE
      RETURN
      END
```

Annexes V.

```
'procedure' EBERLN (A,S,n,in,nbmax,eps,ef,tausq);
'value' n,in,nbmax,eps,ef;
'array' A,S;  'integer' n,in,nbmax;  'real' eps,ef,tausq;
'begin'
'real' c,d,e,cc,ss,sig,cot,co,si,h,g,hj,s1,s2,f,ch,
     sh,c1,c2,a',a2;
'integer' ii,nb,dr,di,k,l,i,j;'for' i:= 1 'step' 1 'until' n 'do'
 'begin' S[i,i]:= 1;
       'for' j:= i + 1 'step' 1 'until' n 'do'
        S[i,j]:= S[j,i]:= 0
 'end';
'for' ii:= 1 'step' 1 'until' in 'do'
'begin'
eps:= eps/ef;
'for' nb:= 0,nb+1 'while' nb < nbmax 'and' dr + di > 0 'do'
 'begin'
dr:= di:= 0;
'for' i:= 2 'step' 1 'until' n 'do'
'for' j:= 1 'step' 1 'until' i-1 'do'
 'begin' c:= A[i,j] + A[j,i];
       d:= A[i,i] - A[j,j];
       'if' abs(c) < eps 'then'
        'begin' cc:= 1;
              ss:= 0
        'end' 'else'
        'begin' cc:= d/c;
              sig:= 'if' cc = 0 'then' 1 'else' sign(cc);
              cot:= cc + sig * sqrt(1 + cc * cc);
              ss:= sig/sqrt( 1 + cot * cot);
              cc:= ss * cot;
              dr:= dr + 1
```

```
   'end';
  e:= A[i,j] - A[j,i];
  ch:= 1;  sh:= 0;
  'if' abs(e) 'ge' eps 'then'
   'begin' co:= cc * cc - ss * ss;
           si:= 2 * ss * cc;
          h:= g:= hj:= 0;
          'for' k:= 1 'step' 1 'until' n 'do'
           'if' k # i 'and' k # j 'then'

 'begin' h:= h + A[i,k] * A[j,k] -
       A[k,i] * A[k,j];
  s1:= A[i,k] ↑ 2 + A[k,j] ↑ 2;
  s2:= A[j,k] ↑ 2 + A[k,i] ↑ 2;
  g:= g + s1 + S2;
  hj:= hj + S1 - s2;
          'end';
          d:= d * co + c * si;
          h:= 2 * h * co - hj * si;
          f:= (2 * e * d - h)/(4 *
             (e * e + d * d) + 2 * g);
'if' abs(f) 'ge' eps 'then'
          'begin' ch:= 1/sqrt(1 - f * f);
  sh:= f * ch;
  di:= di + 1
   'end'
'end';
  c1:= ch * cc - sh * ss;
  c2:= ch * cc + sh * ss;
  s1:= ch * ss + sh * cc;
  s2:= sh * cc - ch * ss;
  'if' abs(s1) + abs(s2) # 0 'then'
'begin' 'for' l:= 1 'step'  'until' n 'do'
'begin' a1:= A[l,i];
```

```
        a2:= A[l,j];
        A[l,i]:= c2 * a1 - s2 * a2;
        A[l,j]:= c  * a2 - S1 * a1;
        a1:= S[l,i];
        a2:= S[l,j];
        S[l,i]:= c2 * a1 - s2 * a2;
        S[l,j]:= c  * a2 - S1 * a1;
        'end';
        'foR' L:= 1 'steP' 1 'uNtil' n 'do'
       'begin' a1:= A[i,l];
        a2:= A[j,l];
        A[i,l]:= c1 * a1 + s1 * a2;
        A[j,l]:= c2 * a2 + s2 * a1;
      'end';
    'end';
  'end';
 'end';
'end';
tausq:= 0;


'for' i:= 1 'step' 1 'until' n 'do'
 'for' j:= i + 1 'step' 1 'until' n 'do'
  tausq:= tausq + A[i,j] ↑ 2 + A[j,i] ↑ 2;
'end';
```

```
'procedure'sgz(S,g,z,a,n);'value'n;'real'S;'integer'n;
        'array'g,z,a;
'begin''integer'i,j,t,nn,k,w5,w3,w7,w6,w9,ii,jj;
'integer'w2,w4,w8,w10,w11,w12;
 'array' df,ddf[1:n],
        t11,t12,t13,t14,t21,t22,t23,t24,th,
    te,tte,tf1,tet[1:12],b3,b4[1:4],
              ta,tb[1:8,1:8],f[1:12],p11,p12,p13,p14,p21,p22,p23,p24[1:12],pk[1:4],pv[1:
8,1:4],thh[1:12];
'real' mutte,mutf1,dtt,dmt;
      'real'kyy,ky,ti,ki,mi,yi,xi,zi,bi,ai,kk,tt,mm,yy,zz,xx,ma;
S:=0;
     'for'i:=1'step'1'until'n'do'
          g[i]:=0;
     t:=n*(n+1)/2;
   'for'i:=1'step'1'until't'do'
          z[i]:=0;
      'for'i:=1'step'1'until'w'do'
'begin'  w4:=i+1+w*4;
        w8:=i+1+w*8;
        w10:=i+1+w*10;
        w11:=i+1+w*11;
        w12:=i+1+w*12;
th[i]:=th[i+4]:=th[i+6]:=th[i+8]:=th[i+10]:=koef1*a[w4];
th[i+2]:=koef1*((1+koef4)*a[w4]);
thh[i]:=thh[i+4]:=thh[i+6]:=thh[i+8]:=thh[i+10]:=a[w4];
thh[i+2]:=a[w4]*(1+koef4);
te[i]:=te[i+2]:=te[i+6]:=te[i+8]:=te[i+10]:=a[w8];
te[i+4]:=a[w8]*(1+koef5);
tte[i]:=tte[i+2]:=tte[i+8]:=tte[i+10]:=tte[i+4]:=a[w10];
tte[i+6]:=a[w10]*(1+koef6);
tf1[i]:=tf1[i+2]:=tf1[i+4]:=tf1[i+6]:=tf1[i+10]:=a[w11];
tf1[i+8]:=a[w11]*(1+koef7);
tet[i]:=tet[i+2]:=tet[i+4]:=tet[i+6]:=tet[i+8]:=a[w12];
tet[i+10]:=a[w12]*(1+koef8);
'end';
     'for'i:=1'step'1'until'12'do'
'begin' kk:=sin(tte[i]);
        tt:=sin(tf1[i]);
        mm:=cos(tf1[i]);
```

```
          yy:=cos(tte[i]);
          xx:=3*te[i];
          zz:=th[i]*yy;
ta[1,1]:=ta[5,5]:=xx-1.5*zz;
ta[1,3]:=ta[5,7]:=ta[2,4]:=ta[6,8]:=ta[3,1]:=ta[7,5]:=ta[4,2]:=ta[8,6]:=
     1.732*tet[i]*te[i];
ta[2,2]:=ta[6,6]:=-xx-0.5*zz;
ta[2,3]:=ta[6,7]:=ta[3,2]:=ta[7,6]:=-th[i]*kk*mm;
ta[ ,2]:=ta[5,6]:=ta[3,4]:=ta[7,8]:=ta[2,1]:=ta[6,5]:=ta[4,3]:=ta[8,7]:=
ta[2,3]*0.866;
ta[3,3]:=ta[7,7]:=-xx+0.5*zz;
ta[4,4]:=ta[8,8]:=xx+1.5*zz;
ta[6,3]:=ta[3,6]:=th[i]*kk*tt;
ta[2,7]:=ta[7,2]:=-ta[6,3];
ta[5,2]:= a[2,5]:=ta[7,4]:=ta[4,7]:=ta[6,3]*0.866;
ta[1,6]:=ta[6,1]:=ta[3,8]:=ta[8,3]:=-ta[5,2];
ta[1,4]:=ta[5,8]:=ta[4,1]:=ta[8,5]:=ta[5,1]:=ta[5,3]:=
ta[5,4]:=ta[6,2]:=ta[6,4]:=ta[7,1]:=ta[7,3]:=ta[8,1]:=
ta[8,2]:=ta[8,4]:=ta[1,5]:=ta[1,7]:=ta[1,8]:=ta[2,6]:=
ta[2,8]:=ta[3,5]:=ta[3,7]:=
ta[4,5]:=ta[4,6]:=ta[4,8]:=0;
 EBERLN(ta,tb,8,3,25,1&-5,10,dk);
        k:=1;
       'for'ii:=1'step'1'until'7'do'
          'for'jj:=2'step'1'until'8'do'
'begin' 'if' abs(ta[ii,ii]-ta[jj,jj])<koef3
       'and'  jj>ii   'then'
'begin'    pk[k]:=ta[ii,ii];
      'for'nn:=1'step'1'until'8'do'
            pv[nn,k]:=tb[nn,ii];
      k:=k+1;
'end';
'end';
  kk:=koef2*0.5*thh[i];
t11[i]:=pk[1]+kk;
t12[i]:=pk[1]-kk;
t13[i]:=pk[2]+kk;
t14[i]:=pk[2]-kk;
t21[i]:=pk[3]+kk;
t22[i]:=pk[3]-kk;
t23[i]:=pk[4]+kk;
t24[i]:=pk[4]-kk;
```

```
    mutte:=tte[i];
    mutfi:=tfi[i];
wkal: vektor(pv, mutte, mutfi, b4, b3);
p11[i]:=b3[1]*kff ;
p12[i]:=b4[1]*kff2;
p13[i]:=b3[2]*kff3;
p14[i]:=b4[2]*kff4;
p21[i]:=b3[3]*kff5;
p22[i]:=b4[3]*kff6;
p23[i]:=b3[4]*kff7;
p24[i]:=b4[4]*kff8;
'end';
     'for' t:=6 'step' 1 'until' cis-5 'do'
'begin'
beta[t]:=0;
gamma[t]:=0;
ma:=a[1]+a[n]*x[t];
          'for'i:=1'step'1'until'w'do'
'begin' w5:=1+i+5*w;
        w7:=1+i+7*w;
        w9:=1+i+9*w;
       w3:=1+i+w*3;
      ky:=a[i+w+1]↑2/4;
  kk:=x[t]+kfi*a[w5]-a[w7]+a[w9];
  tt:=x[t]+kfe*a[w5]-a[w7]-a[w9];
  mm:=x[t]+kff*a[w5]-a[w7]-a[w9];
  yy:=x[t]-kff*a[w5]-a[w7]-a[w9];
  xx:=x[t]-kfe*a[w5]-a[w7]-a[w9];
  zz:=x[t]-kfi*a[w5]-a[w7]+a[w9];
  ki:=kk↑2+ky;
  ti:=tt↑2+ky;
  mi:=mm↑2+ky;
  yi:=yy↑2+ky;
  xi:=xx↑2+ky;
  zi:=zz↑2+ky;
  bi:=ma*a[w3];
  df[w3]:=sza/ki+szb/ti+szc/mi+szd/yi+sze/xi+szf/zi;
  df[i+w+1]:=bi*a[i+w+1]/2*(sza/ki↑2+szb/ti↑2+szc/mi↑2+
                          szd/yi↑2+sze/xi↑2+szf/zi↑2);
```

```
  beta[t]:=beta[t]+a[w3]*df[w3];
    kmk[t,1]:=beta[t];
   df[w3]:=-ma*df[w3];
kk:=kk/ki↑2*sza;
tt:=tt/ti↑2*szb;
mm:=mm/mi↑2*szc;
yy:=yy/yi↑2*szd;
xx:=xx/xi↑2*sze;
zz:=zz/zi↑2*szf;
 df[w5]:=-2*bi*(-kf1*kk-kfe*tt-kff*mm
                +kff*yY+kfe*xx+kf1*zz);
 df[w7]:=-2*bi*(kk+tt+mm+yy+xx+zz);
 df[w9]:=2*bi*(kk-tt-mm-yy-xx+zz);
    w2:=1+i+w*2;
  w6:=1+i+w*6;
  ky:=a[i+1]↑2/4;
     ti:=x[t]-a[w6]-t11[i];
     ki:=x[t]-a[w6]-t12[i];
     mi:=x[t]-a[w6]-t13[i];
     yi:=x[t]-a[w6]-t14[i];
     xi:=x[t]-a[w6]-t21[i];
     zi:=x[t]-a[w6]-t22[i];
     bi:=x[t]-a[w6]-t23[i];
     ai:=x[t]-a[w6]-t24[i];
  tt:=ti↑2+ky;
  kk:=ki↑2+ky;
  mm:=mi↑2+ky;
  yy:=yi↑2+ky;
  xx:=xi↑2+ky;
  zz:=zi↑2+ky;
mutte:=bi↑2+ky;
mutfi:=ai↑2+ky;
 df[w2]:=-ma*(p11[i]/tt+p12[i]/kk+p13[i]/mm+p14[i]/yy+
              p21[i]/xx+p22[i]/zz+p23[i]/mutte+p24[i]/mutfi);
 df[w6]:=-2*ma*a[w2]*(p11[i]*ti/tt↑2+p12[i]*ki/kk↑2+
                      p13[i]*mi/mm↑2+p14[i]*yi/yy↑2+
                      p21[i]*xi/xx↑2+p22[i]*zi/zz↑2+
                      p23[i]*bi/mutte↑2+p24[i]*ai/mutfi↑2);
```

```
  df[i+1]:=a[i+1]/2*ma*a[w2]*(
p11[i]/tt↑2+p12[i]/kk↑2+p13[i]/mm↑2+p14[i]/yy↑2+
p21[i]/xx↑2+p22[i]/zz↑2+p23[i]/mutte↑2+p24[i]/mutfi↑2);
  gamma[t]:=gamma[t]+a[w2]*(-df[w2])/ma;
'end';
beta[t]:=beta[t]+gamma[t];
       df[1]:=1-beta[t];
       df[n]:=x[t]*df[1];
  beta[t]:=ma*df[1];
          'for'j:=0'step'2'until'10'do'
            'for'i:=1'step'1'until'w'do'
'begin'     ky:=a[i+1]↑2/4;
           w6:=1+i+w*6;
      f[i+j]:=a[i+1+w*2]*(p11[i+j]/((x[t]-a[w6]-t11[i+j])↑2+kY)+
                        p12[i+j]/((x[t]-a[w6]-t12[i+j])↑2+ky)+
                        p13[i+j]/((x[t]-a[w6]-t13[i+j])↑2+ky)+
                        p14[i+j]/((x[t]-a[w6]-t14[i+j])↑2+ky)+
                        p21[i+j]/((x[t]-a[w6]-t21[i+j])↑2+ky)+
                        p22[i+j]/((x[t]-a[w6]-t22[i+j])↑2+ky)+
                        p23[i+j]/((x[t]-a[w6]-t23[i+j])↑2+ky)+
                        p24[i+j]/((x[t]-a[w6]-t24[i+j])↑2+ky));
'end';
```

```
'for'i:=1'step'1'until'w'do'
'begin' df[i+1+w*4]:=ma*(f[i]-f[i+2])/(a[i+1+w*4]*koef4);
  df[i+1+w*8]:=ma*(f[i]-f[i+4])/(a[i+1+w*8]*koef5);
  df[i+1+w*10]:=ma*(f[i]-f[i+6])/(a[i+1+w*10]*koef6);
    df[i+1+w*11]:=ma*(f[i]-f[i+8])/(a[i+1+w*11]*koef7);
  df[i+1+w*12]:=ma*(f[i]-f[i+10])/(a[i+1+w*12]*koef8);
'end';
    'for'i:=1'step'1'until'n'do'
          ddf[i]:=df[i];
 'if' eta[1]=111'then' 'goto' cim;
   'if'eta[1]=222 'then'
'begin' ddf[2]:=df[2]+df[4]+df[5];
     'for'nn:=2'step'1'until'3'do'
'begin' ddf[nn+2]:=0;
          a[nn+2]:=a[2];
          l[nn+2]:=0;
'end';
'end';
     'if' eta[5]=333'then'
'begin' ddf[6]:=df[6]+df[8]+df[9];
        'for'nn:=2'step'1'until'3'do'
'begin' ddf[nn+6]:=0;
          a[nn+6]:=a[6];
          l[nn+6]:=0;
'end';
'end';
      beta[t]:=gamma[t]:=0;
'for' i:=1'step'1'until'w'do'
'begin' w5:=i+1+w*5;
        w3:=1+i+3*w;
        W7:=1+i+7*W;
        w9:=1+i+9*w;
         w6:=1+i+w*6;
      ky:=a[i+w+1]↑2/4;
beta[t]:=beta[t]+(a[w3]*(sza/((x[t]+kfi*a[w5]-a[w7]+a[w9])↑2+ky)+
                           szb/((x[t]+kfe*a[w5]-a[w7]-a[w9])↑2+ky)+
```

```
                                         szc/((x[t]+kff*a[w5]-a[w7]-a[w9])↑2+ky)+
                                         szd/((x[t]-kff*a[w5]-a[w7]-a[w9])↑2+kY)+
                                         sze/((x[t]-kfe*a[w5]-a[w7]-a[w9])↑2+ky)+
                                         szf/((x[t]-kfi*a[w5]-a[w7]+a[w9])↑2+ky)));
   kyy:=a[i+1]↑2/4;
   gamma[t]:=gamma[t]+(a[i+1+w*2]*
               ((p11[i]/((x[t]-a[W6]-t11[i])↑2+kyy))+
                         (p12[i]/((x[t]-a[w6]-t12[i])↑2+kyy))+
                         (p13[i]/((x[t]-a[w6]-t13[i])↑2+kyy))+
                         (p14[i]/((x[t]-a[w6]-t14[i])↑2+kyy))+
                         (p21[i]/((X[t]-a[w6]-t21[i])↑2+kyy))+
                         (p22[i]/((X[t]-a[w6]-t22[i])↑2+kyy))+
                         (p23[i]/((x[t]-a[w6]-t23[i])↑2+kyy))+
                         (p24[i]/((x[t]-a[w6]-t24[i])↑2+kyy))));
'end';
 beta[t]:=ma*(1-beta[t]-gamma[t]);
cim:  'if' y[t]=b*65535 'or' al[t]=1
   'then'
'begin' y[t]:=beta[t];
        al[t]:=1;
'end';
  dtt:=sqrt(y[t]);
  dmt:=beta[t]-y[t];
      form(z,g,ddf,dmt,dtt,n);
      S:=S+(beta[t]-y[t])↑2/y[t];
'end';
'end';
```

```
      SUBROUTINE VEKTOR (VK,RTE,RFI,B3,B4)
      COMPLEX V,CA21,CA22,CEXPFI
      DIMENSION V(4,4),VK(8,4),B3(4),B4(4)
      Q=0.57735026919
      CEXPFI=CMPLX(COS(RFI),-SIN(RFI))
      F=0.5*RTE
      CA11=SIN(F)
      CA12=COS(F)
      DO 15 I=1,4
      DO 15 J=1,4
15    V(I,J)=CMPLX(VK(I,J),VK(I+4,J))
      CA21=-CA12*CEXPFI
      CA22=+CA11*CEXPFI
      DO 17 J=1,4
     OB3(J)=((CABS(CA11*V(1,J)+Q*CA11*V(3,J)
     1+CA21*V(4,J)+Q*CA21*V(2,J)))**2
     2+(CABS(CA11*V(1,J)-Q*CA11*V(3,J)
     3-CA21*V(4,J)+Q*CA21*V(2,J)))**2
     4+(CABS(CA11*V(2,J)+CA21*V(3,J)))**2*1.3333333333)*4.1887902048
     OB4(J)=((CABS(CA12*V(1,J)+Q*CA12*V(3,J)
     1+CA22*V(4,J)+Q*CA22*V(2,J)))**2
     2+(CABS(CA12*V(1,J)-Q*CA12*V(3,J)
     3-CA22*V(4,J)+Q*CA22*V(2,J)))**2
     4+(CABS(CA12*V(2,J)+CA22*V(3,J)))**2*1.3333333333)*4.1887902048
17    CONTINUE
      RETURN
      END
```

Annexes VI.

```
'begin' 'integer'n,nn;
      nn:=read;
       n:=read;
   newline(3);space(50);
   writetext('('program%jgja')');newline(3);
'begin' 'array's,x,y,z,x1,y1,z1[1:n],f[1:9],u[1:3],
               ta,tb[1:3,1:3],x2,y2,z2,s2[1:n,1:nn],mm[1:11,1:nn];
       'integer'i,d,w,k,t,j,m,xa,xb,jj;
'real'aa,ba,ca,da,a,b,c,r,r2,r3,r5,q,p,
     u1,v,l1,l2,l3,ka,kb,kc,a1,b1,c1;
'procedure' jakobi(n,rho,a,s);
       'value'n,rho;
'integer'n;
'real'rho;
'array'a,s;
'begin''real' norm1,norm2,thr,mu,omega,sint,cost,int1,v1,v2,v3;
      'integer'i,j,p,q,ind;
       'for'i:=1'step'1'until'n'do'
             'for'j:=1'step'1'until'i'do'
    'if'i=j'then's[i,j]:=1'else's[i,j]:=s[j,i]:=0;
  int1:=0;
      'for'i:=2'step'1'until'n'do'
  'for'j:=1'step'1'until'i-1'do'
       int1:=int1+2*a[i,j]↑2;
norm1:=thr:=sqrt(int1);
norm2:=(rho/n)*norm1;
  ind:=0;
 main:  thr:=thr/n;
 main1:  'for'q:=2'step'1'until'n'do'
            'for'p:=1'step'1'until'q-1'do'
     'if' abs(a[p,q])>thr 'then'
'begin'  ind:=1;
      v1:=a[p,p];
      v2:=a[p,q];
      v3:=a[q,q];
```

```
        mu:=0.5*(v1-v3);
        omega:='if' mu=0'then'-1'else'-sign(mu)*v2/sqrt(v2↑2+mu↑2);
        sint:=omega/sqrt(2*(1+sqrt(1-omega↑2)));
        cost:=sqrt(1-sint↑2);
      'for'i:=1'step'1'until'n'do'
'begin'  int1:=a[i,p]*cost-a[i,q]*sint;
         a[i,q]:=a[i,p]*sint+a[i,q]*cost;
         a[i,p]:=int1;
         int1:=s[i,p]*cost-s[i,q]*sint;
         s[i,q]:=s[i,p]*sint+s[i,q]*cost;
         s[i,p]:=int1;
'end';
      'for'i:=1'step'1'until'n'do'
'begin'  a[p,i]:=a[i,p];
         a[q,i]:=a[i,q];
'end';
  a[p,p]:=v1*cost↑2+v3*sint↑2-2*v2*sint*cost;
  a[q,q]:=v1*sint↑2+v3*cost↑2+2*v2*sint*cost;
  a[p,q]:=a[q,p]:=(v1-v3)*sint*cost+v2*(cost↑2-sint↑2);
'end';
  'if' ind=1'then'
'begin'  ind:=0;
    'go to' main1
'end';
'if' th>norm2'then' 'go to' main;
'end';
     'for'j:=1'step'1'until'nn'do'
'begin'      'for'i:=1'step'1'until'11'do'
                 mm[i,j]:=read;
             'for'i:=1'step'1'until'n'do'
     'begin'    x2[i,j]:=read;
                y2[i,j]:=read;
                z2[i,j]:=read;
                s2[i,j]:=read;
     'end';
```

```
 'end';
          'for'jj:=1'step'1'until'nn'do'
 'begin'   xa:=mm[1,jj];
           xb:=mm[2,jj];
            w:=mm[3,jj];
            a:=mm[4,jj];
            b:=mm[5,jj];
            c:=mm[6,jj];
            v:=mm[7,jj];
           l1:=mm[8,jj];
           l2:=mm[9,jj];
           l3:=mm[10,jj];
            m:=mm[11,jj];
          'for'i:=1'step'1'until'n'do'
'begin' x1[i]:=x2[i,jj];
        y1[i]:=y2[i,jj];
        z1[i]:=z2[i,jj];
         s[i]:=s2[i,jj];
'end';
  newline(3);
  space(30);writetext('('nomer%serii')');
  space(3);print(xb,4,0);newline(2);
  space(30);writetext('('nulevaja%tocska')');
  space(3);print(w,3,0);
  newline(2);
  ka:=0;
  kb:=0;
  kc:=0;
  space(20);writetext('('x')');
  space(20);writetext('('y')');
  space(20);writetext('('z')');
  space(20); writetext('('s')');
newline(2);
    'for' i:=1 'step' 1 'until' n 'do'
'begin' space(15); print(x1[i],2,5);
        space(12); print(y1[i],2,5);
```

```
        space(12); print(z1[i],2,5);
        space(12); print(s[i],2,5);
        newline(1);
   'end';
        'for'd:=1'step'1'until'xa'do'
     'begin' newline(2);
       space(10); writetext('('sag')');
       space(3); print(xa,2,0);
       space(3); writetext('('ka=')');
       print(ka,2,6); space(3); writetext('('kb=')');
       print(kb,2,6); space(3); writetext('('kc=')');
       print(kc,2,6); newline(1);
          a:=a+ka;
          b:=b+kb;
          c:=c+kc;
      'for'i:=1'step'1'until'n'do'
'begin'  x[i]:=x1[i];
          y[i]:=y1[i];
          z[i]:=z1[i];
 end';
       a1:=x[w];
       b1:=y[w];
       c1:=z[w];
'for'i:=1'step'1'until'n'do'
'begin'x[i]:=(x[i]-a1)*a;
        y[i]:=(y[i]-b1)*b;
        z[i]:=(z[i]-c1)*c;
'end';
  'for'i:=1'step'1'until'9'do'
       f[i]:=0;
       'for'k:=-m'step'1'until'm'do'
          'for't:=-m'step'1'until'm'do'
            'for'i:=-m'step'1'until'm'do'
               'for'j:=1'step'1'until'n'do'
'begin' 'if'k=0'and't=0'and'i=0'and'j=w 'then'
          'go to' mk;
```

```
  r2:=(x[j]+a*i)↑2+(y[j]+b*t)↑2+(z[j]+c*k)↑2;
  r3:=r2*sqrt(r2);
  r5:=r3*r2;
  f[1]:=f[1]+((3*(x[j]+a*i)↑2/r2-1)*s[j]/r3);
  f[2]:=f[2]+(3*s[j]*(x[j]+a*i)*(y[j]+b*t)/r5);
  f[3]:=f[3]+(3*s[j]*(x[j]+a*i)*(z[j]+c*k)/r5);
  f[4]:=f[2];
  f[5]:=f[5]+((3*(y[j]+b*t)↑2/r2-1)*s[j]/r3);
  f[6]:=f[6]+(3*s[j]*(y[j]+b*t)*(z[j]+c*k)/r5);
  f[7]:=f[3];
  f[8]:=f[6];
  f[9]:=f[9]+((3*(z[j]+c*k)↑2/r2-1)*s[j]/r3);
mk:  'end';
  newline(4);
    'for'i:=1'step'1'until'9'do'
'begin' print(f[i],0,8); newline(1);
'end';      k:=0;
    'for'i:=1'step'1'until'3'do'
'begin'  'for'j:=1'step'1'until'3'do'
         ta[i,j]:=f[j+k];
         k:=k+3;
'end';
        jakobi(3,1&-8,ta,tb);
       u[1]:=ta[1,1];
       u[2]:=ta[2,2];
       u[3]:=ta[3,3];
   newline(1);
    writetext('('vektor')');  newline(1);
  'for'i:=1'step'1'until'3'do'
    'for'j:=1'step'1'until'3'do'
'begin' space(3);
        print(tb[j,i],8,8);
   newline(1);
'end';
    space(2);  writetext('('korni')');
   newline(2);
  'for'i:=1'step'1'until'3'do'
```

```
'begin'   print(U[i],0,8);
   newline(1);
'end';
     'for'i:=1'step'1'until'2'do'
'begin'  u1:=u[i];
      t:=i;
    'for'j:=i+1'step' 1'until'3'do'
'if'  abs(u[j])>abs(u1) 'then'
'begin'  u1:=u[j];
      t:=j;
'end';
     'if'  t#i 'then'
'begin'  u[t]:=u[i];
         u[i]:=u1;
'end';
'end';
      aa:=(u[3]-u[2])/u[1];
     ba:=u[1];
     ca:=v*ba*(1+aa↑2/3)↑0.5;
     newline(1);
     space(10);
  writetext('('kvadrupolnoe%raszseplenie')');
  space(10);
  writetext('('gradient')');
 space(10);
  writetext('('parametr%aszsimetrii')');
  space(10);
  writetext('('parametr%resetki')');
    newline(1);
  space(20);print(ca,1,3);
  space(10); print(ba,3,5);
  space(8); print(aa,3,5);
  space(10); print(a,2,5);
  space(3); print(b,2,5);
  space(3); print(c,2,5);
```

```
    ka:=11;
    kb:=12;
    kc:=13;
  newline(2);
'end';
'end';
      writetext('('konec')');
'end';
'end';
```

Annexes VII.

```
      MASTER JGGR
      INTEGER D,K,T
      REAL L1,L2,L3
      DIMENSION S(100),X(100),Y(100),Z(100),X1(100),Y1(100),Z1(100),
     CF(9),U(3),TA(3,3),TB(3,3)
      READ (11,5)N,NA,NW,A,B,C,V,L1,L2,L3,M
5     FORMAT(I0,I0,I0,F0.0,F0.0,F0.0,F0.0,F0.0,F0.0,F0.0,I0,)
      READ (11,6)(X1(I),Y1(I),Z1(I),S(I),I=1,N)
6     FORMAT(F0.0,F0.0,F0.0,F0.0,)
      WRITE(10,7)
7     FORMAT(1H0,)
      KA=0
      KB=0
      KC=0
      DO 10 D=1,NA,1
      A=A+KA
      B=B+KB
      C=C+KC
      DO 11 I=1,N,1
      X(I)=X1(I)
      Y(I)=Y1(I)
11    Z(I)=Z1(I)
      A1=X(NW)
      B1=Y(NW)
      C1=Z(NW)
      DO12 I=1,N,1
      X(I)=(X(I)-A1)*A
      Y(I)=(Y(I)-B1)*B
12    Z(I)=(Z(I)-C1)*C
      DO 13 I=1,9,1
13    F(I)=0
      L=2*M+1
      DO 14 K=1,L
      DO 14 T=1,L
      DO 14 I=1,L
      DO 14 J=1,N
      IF(K-M-1) 18,15,18
15    IF(T-M-1) 18,16,18
16    IF(I-M-1) 18,17,18
17    IF(J-NW)18,100,18
18    R2=(X(J)+A*(I-M-1))**2+(Y(J)+B*(T-M-1))**2+(Z(J)+C*(K-M-1))**2
      R3=R2*SQRT(R2)
      R5=R2*R3
      F(1)=F(1)+((3*(X(J)+A*(I-M-1))**2/R2-1)*S(J)/R3)
      F(2)=F(2)+(3*S(J)*(X(J)+A*(I-M-1))*(Y(J)+B*(T-M-1))/R5)
      F(3)=F(3)+(3*S(J)*(X(J)+A*(I-M-1))*(Z(J)+C*(K-M-1))/R5)
      F(4)=F(2)
      F(5)=F(5)+((3*(Y(J)+B*(T-M-1))**2/R2-1)*S(J)/R3)
      F(6)=F(6)+(3*S(J)*(Y(J)+B*(T-M-1))*(Z(J)+C*(K-M-1))/R5)
      F(7)=F(3)
      F(8)=F(6)
      F(9)=F(9)+((3*(Z(J)+C*(K-M-1))**2/R2-1)*S(J)/R3)
100   MLL= 0
```

```
14     CONTINUE
       WRITE(10,50)NW
50     FORMAT(//,30X,12HPROGRAM JGGR,//,15X,15HNULEVAJA TOCSKA,2X,3HNW=,
      CI2,///,)
       WRITE(10,51)(F(I),I= ,9)
51     FORMAT(4X,E15.8)
       K=0
       DO 19 I=1,3,1
       DO 20 J=1,3,1
20     TA(I,J)=F(J+K)
       K=K+3
19     CONTINUE
       CALL JAKOBG(3,0.00000001,TA,TB)
       U( )=TA(1,1)
       U(2)=TA(2,2)
       U(3)=TA(3,3)
       WRITE( 0,70)
70     FORMAT(/,3X,6HVEKTOR,/,)
       DO 71 J=1,3
       WRITE(10,72)(TB(I,J),I=1,3)
72     FORMAT(/,3X,F15.8,)
71     CONTINUE
       WRITE(10,73)
73     FORMAT(/,3X,5HKORNI,/,)
       WRITE(10,53)(U(I),I= ,3)
53     FORMAT(/,3X,F15.8,)
       DO 25 I=1,2,1
       U1=U(I)
       T=I
       DO 27 J=I+1,3,1
       IF (ABS(U(J)-ABS(U1))) 27,27,28
28     U1=U(J)
       T=J
27     CONTINUE
       IF(T-I)29,120,29
29     U(T)=U(I)
       U(I)=U1
 20    CONTINUE
25     CONTINUE
       AA=(U(3)-U(2))/U(1)
       BA=U(1)
       CA=V*BA*(1+AA**2/3)**0.5
       WRITE(10,55) CA,BA,AA,A,B,C
55     FORMAT(/,10X,25HKVADRUPOLNOE RASZSEPLENIE,5X,8HGRADIENT,10X,
      C20HPARAMETER ASZIMETRII,10X,17HPARAMETER RESETKI,/,20X,F8.5,10X,
      CF8.5,8X,F8.5,10X,F7.5,3X,F7.5,3X,F7.5,/,)
       KA=L1
       KB=L2
       KC=L3
10     CONTINUE
       WRITE(10,56)
56     FORMAT(//,10X,5HKONEC,/,)
       STOP
       END
```

```
      SUBROUTINE JAKOBG (N,RHO,A,S)
      REAL MU
      DIMENSION S(N,N),A(N,N)
      DO 10 I=1,N
      DO 10 J=1,I
      IF (I-J)  2,14,12
14    S(I,J)=
      GO TO 11
12    S(I,J)=0
      S(J,I)=0
11    CONTINUE
10    CONTINUE
      XINT1=0
      DO 15 I=2,N
      II=I-1
      DO 15 J=1,II
      XINT1=XINT1+2*A(I,J)**2
15    CONTINUE
      XNORM1=SQRT(XINT1)
      THR=SQRT(XINT1)
      XNORM2=(RHO/N)*XNORM1
      IND=0
16    THR=THR/N
17    DO  8 IQ=2,N
      IIP=IQ-1
      DO 18 IP=1,IIP
      IF(ABS(A(IP,IQ))-THR) 19,19,20
20    IND=1
      V1=A(IP,IP)
      V2=A(IP,IQ)
      V3=A(IQ,IQ)
      MU=0.5*(V1-V3)
      IF (MU-0) 22,21,22
21    OMEGA=-1
      GO TO 23
22    OMEGA=-SIGN(MU)*V2/SQRT(V2**2+MU**2)
23    SINT=OMEGA/SQRT(2*( +SQRT( -OMEGA**2)))
      COST=SQRT(1-SINT**2)
      DO 24 I=1,N
      XINT1=A(I,IP)*COST-A(I,IQ)*SINT
      A(I,IQ)=A(I,IP)*SINT+A(I,IQ)*COST
      A(I,IP)=XINT1
      XINT1=S(I,IP)*COST-S(I,IQ)*SINT
      S(I,IQ)=S(I,IP)*SINT+S(I,IQ)*COST
      S(I,IP)=XINT1
24    CONTINUE
      DO 25 I=1,N
      A(IP,I)=A(I,IP)
      A(IQ,I)=A(I,IQ)
25    CONTINUE
      A(IP,IP)=V1*COST**2+V3*SINT**2-2*V2*SINT*COST
      A(IQ,IQ)=V1*SINT**2+V3*COST**2+2*V2*SINT*COST
      A(IP,IQ)=(V1-V3)*SINT*COST+V2*(COST**2-SINT**2)
      A(IQ,IP)=A(IP,IQ)
19    CONTINUE
18    CONTINUE
      IF (IND-1) 27,26,27
26    IND=0
      GO TO  7
27    IF (THR-XNORM2) 29,29,28
28    GO TO 16
29    KKK=0
      RETURN
      END
```

Printed in the Central Research Institute for Physics, Budapest

Kiadja a KFKI Könyvtár- és Kiadói Osztály. O.v.: Dr.Farkas Istvánné
Szakmai lektor: Pócs Lajos Nyelvi lektor: Kovács Jenőné
Készült a KFKI házi sokszorositójában. F.v.: Gyenes Imre
Példányszám: 130 Munkaszám: 4774 Budapest, 1969. október 14.