MAGASABBFOKÚ EGYENLETEK TÁRGYALÁSÁNAK EGY MÓDJA A SZÁMÍTÁSTECHNIKA ELEMEINEK FELHASZNÁLÁSÁVAL

KISS PÉTER—SZEPESSY BÁLINT

(Közlésre érkezett: 1973. január 15.)

A korszerű matematikaoktatás megkívánja, hogy az oktatás folyamatában a legkorszerűbb eszközöket és módszereket is felhasználjuk. Ezt a célt szolgálja a számítástechnika elemeinek is a felhasználása a matematika különböző fejezeteinek a tárgyalása során. A tanárképző főiskolák matematikai tantervei és programjai lehetőséget biztosítanak erre. Elősegíti ezt az a tény is, hogy a főiskolák számítógépekkel való ellátása is egyre bővül. Intézményünk matematikai tanszéke célul tűzte ki, hogy keresi a számítástechnika felhasználási lehetőségeit a különböző szakterületek oktatásán belül.

Ebben a dolgozatban az „Algebra és számelmélet" című tárgy egyik fejezetét ragadtuk ki és tárgyaljuk az előbb elmondottak szerint. A többi fejezetek hasonló szellemben való tanításának kipróbálása folyamatban van, s ezek későbbi dolgozatok témái lesznek.

Megemlítjük, hogy ezen elképzelések megvalósítása során figyelembe kell vennünk azt a tényt, hogy algebrát hallgatóink az első és a második éven tanulnak, amikor is még nem, illetve minimális számítástechnikai ismeretekkel rendelkeznek.

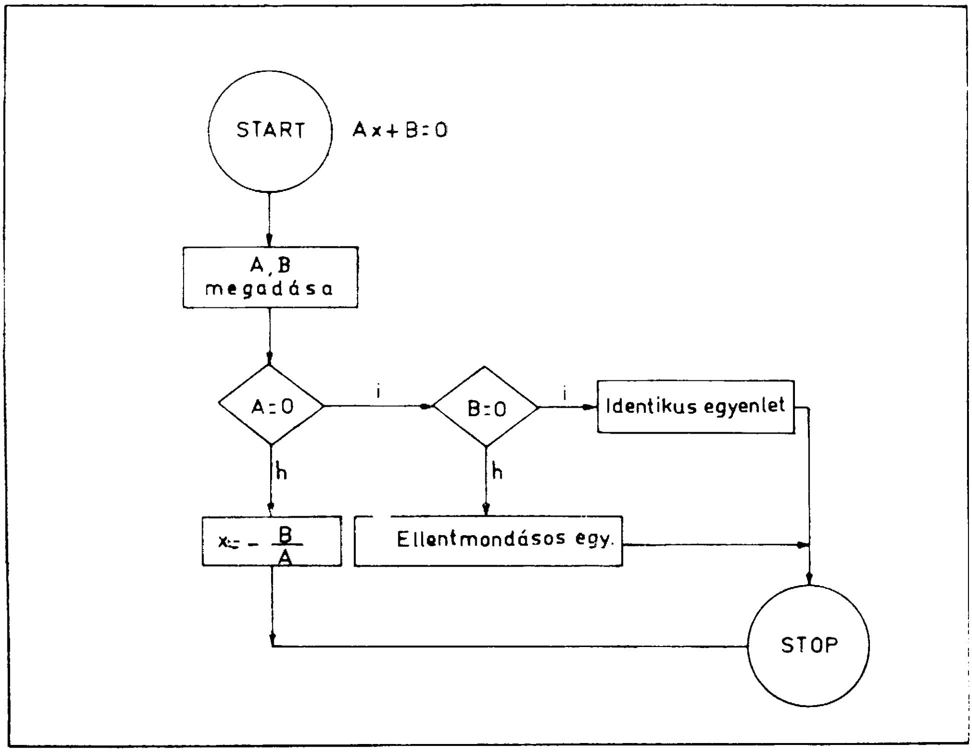
Az egyes eljárások és módszerek lényegének megértését és alkalmazását — a hallgatók körében — megkönnyítik a folyamatábrák alkalmazása. Ezekkel több célt érünk el. Egyrészt könnyebben sajátítják el a tárgyalt anyagot, érthetőbbek lesznek a logikai összefüggések, másrészt logikai készségüket fejlesztjük és hozzászoktatjuk őket a számítástechnikában nélkülözhetetlen algoritmusban való gondolkodáshoz. Azt is elérjük, hogy a számítástechnika elemeinek és egy programnyelvnek a megismerése után ezen folyamatábrák alapján programot írhatnak.

Megemlítjük még, hogy amikor a címben jelzett anyagrészek tárgyalásához érkezünk, a folyamatábra fogalma, a bennük alkalmazott jelölések már nem ismeretlenek a hallgatók számára.

A témával kapcsolatos néhány komplett programot is megadunk, amelyeket elsősorban a gyakorlatokat vezető tanárok használhatnak fel. A programokat ODRA 1204-es számítógépnél használható ALGOL 1204-es programnyelven írtuk, ugyanis ilyen gép működik tanszékünkön. Komplett programok közlését az is indokolja, hogy a kiírási utasításokkal együtt jobban megmutathatjuk azok gyakorlati alkalmazhatóságát.

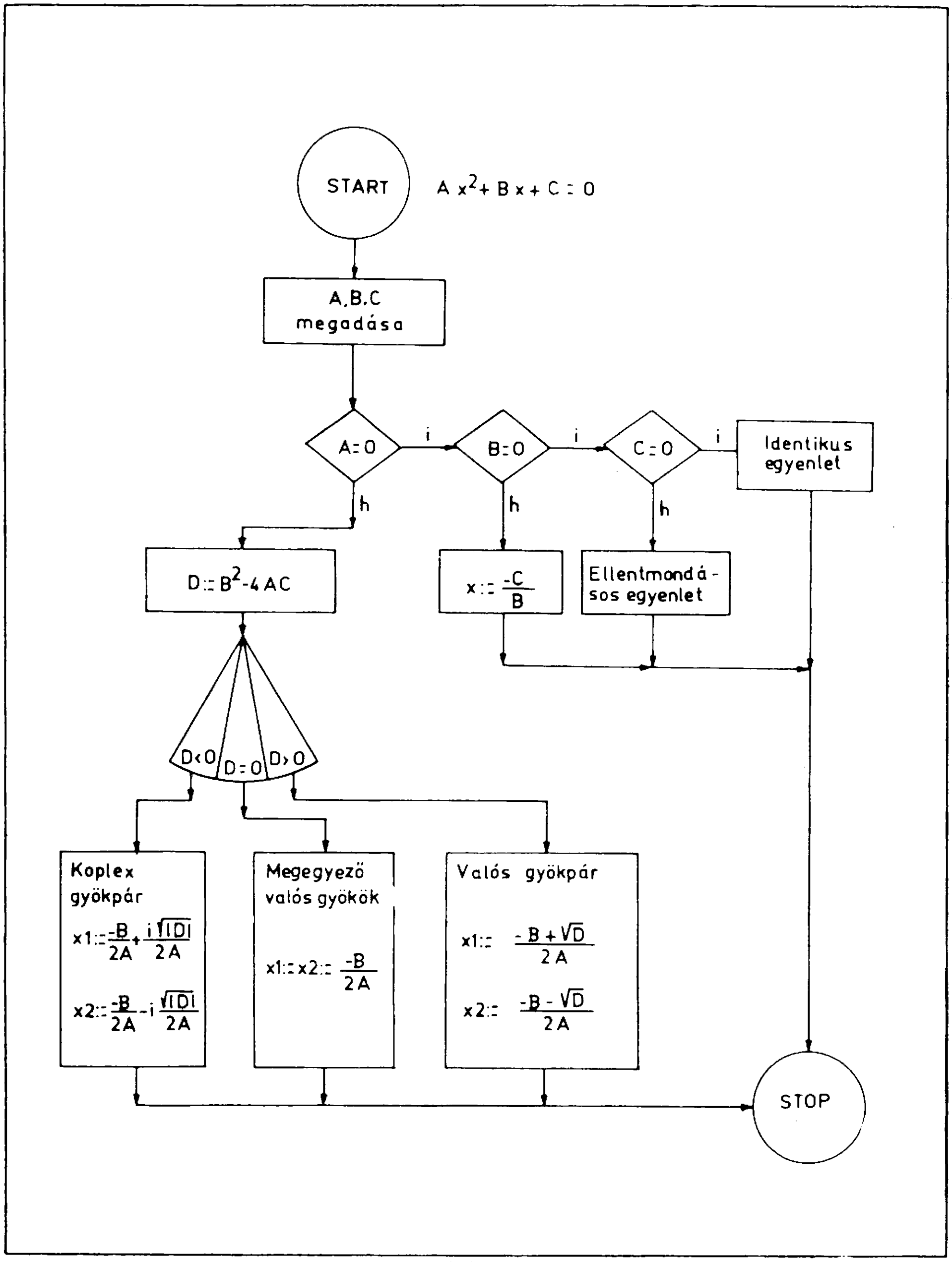
A főiskolai algebraanyag a magasabbfokú egyenletek főcím alatt az első-, másod-, harmad- es negyedfokú algebrai egyenleteket, valamint az egyenletek közelítő megoldásait tárgyalja. Ezt a sorrendet és a szokásos felépítést az alábbiakban igyekszünk változatlanul hagyni. Az egyenletek együtthatóit azonban a bonyolultabb képletek elkerülése, a folyamatábrák áttekinthetősége, valamint nagyobb gyakorlati jelentősége miatt a valós számtestből vesszük.

Az elsőfokú egyenletekkel kapcsolatban az előadásokon némi magyarázat kíséretében — az általános iskolai használhatóságot kihangsúlyozva — csak a folyamatábrát rajzoljuk fel. (1. ábra.)



1.ábra

A másodfokú egyenleteket — egyrészt, mert a középiskolából részben már ismert, másrészt, mert a komplex szám fogalma és az azok körében végzett műveletek megelőzik az egyenletek tárgyalását — önálló feldolgozásra jelöljük ki. Így az előadáson mintegy összefoglalásképpen az algoritmus megbeszélése és a folyamatábra elkészítése nem jelent problémát. (2. ábra.)


START
A x²+ B x + C = 0
A,B,C
megadása
A=0
i
B=0
i
C=0
i
Identikus
egyenlet
h
h
h
D:= B²-4AC
x := -C / B
Ellentmondá-
sos egyenlet
D<0
D=0
D>0
Koplex
gyökpár
x1:= -B/2A + i√|D|/2A
x2:= -B/2A - i √|D|/2A
Megegyező
valós gyökök
x1:=x2:= -B/2A
Valós gyökpár
x1:= (-B + √D)/2A
x2:= (-B - √D)/2A
STOP

2. ábra

Célszerű az első, illetve a második ábrát úgy készíteni, hogy bármely, legfeljebb első- illetve másodfokú egyenletre alkalmazható legyen. Így a második ábra az elsőt tartalmazza.

A valós együtthatós harmadfokú egyenletek folyamatábráját az elmélettel párhuzamosan tárgyalhatjuk.

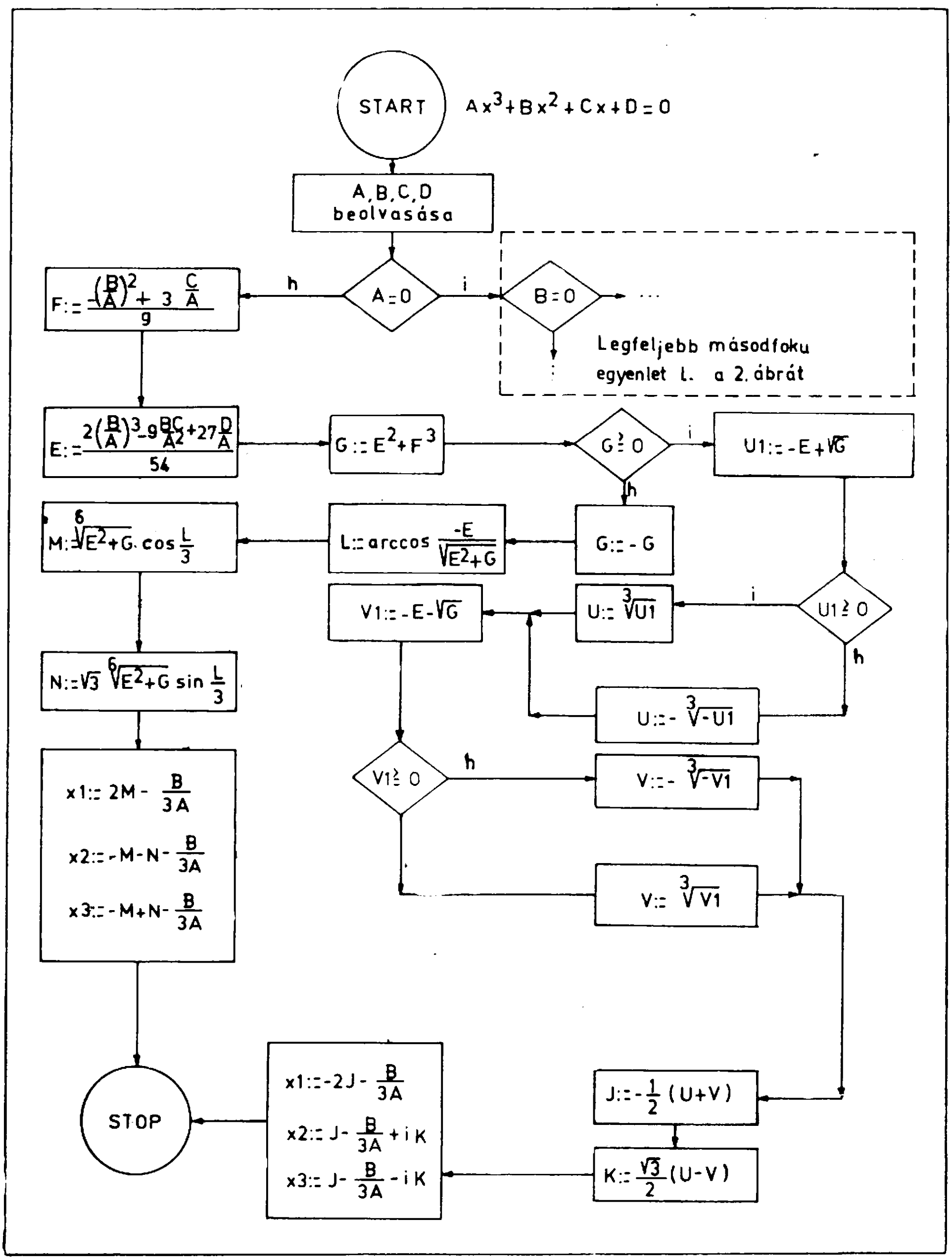
Ha az $ax^3 + bx^2 + cx + d = 0$ egyenlet harmadfokú tagjának az együtthatója nulla ($a = 0$), akkor legfeljebb másodfokú az egyenlet, s ennek megoldási algoritmusa és folyamatábrája már az előzőekben szerepelt. Ha $a \neq 0$, akkor $x = y - \frac{b}{3a}$ ismert helyettesítéssel a másodfokú tagot kiküszöböljük az egyenletből, s az egyszerűbb $y^3 + py + q = 0$ egyenlethez jutunk. A p és q, illetve $\frac{p}{3} = F$ és $\frac{q}{2} = E$ értékek meghatározása után ezeket a folyamatábrán feltüntetjük.

Ha $E^2 + F^3 = G$ nem negatív, akkor a Cardano-féle képletet, az $y_1 = u + v$, $y_2 = -\frac{1}{2}(u+v) + i\frac{\sqrt{3}}{2}(u - v)$, $y_3 = -\frac{1}{2}(u+v) - i\frac{\sqrt{3}}{2}(u - v)$ képleteket, valamint az $x = y - \frac{b}{3a}$ helyettesítést figyelembe véve az egyenlet gyökeit meghatározhatjuk. Ezt az utat a folyamatábrán is végigjárjuk. (3. ábra.) (Megemlítjük, hogy zérus vagy negatív alapnak valós kitevőjű hatványozása $a^x = e^{x \ln a}$ azonosság szerint — s a legtöbb számítógép ezen azonosság szerint számol — nincs értelmezve. Így a $\sqrt[3]{Ul}$ értékét, ha Ul negatív $-\sqrt[3]{-Ul}$ átalakítással számítjuk ki. A 3. ábrán — mivel később ennek alapján programot készítünk, ezt is feltüntetjük.) A valós és a képzetes gyökök számát a folyamatábra alapján kihangsúlyozzuk.

Ha $G < 0$, akkor az úgynevezett casus irreducibilis áll elő. Ebben az esetben ki lehet számítani a gyököket kizárólag a valós számok körében maradva. Mivel $E^2 + F^3 = G < 0$, az ismert képlet szerint $Ul = U^3 = -E + i\sqrt{G}$ írható. Ennek trigonometrikus alakja legyen $r(\cos\varphi + i\sin\varphi)$, és $\bar{G} = -G$. Tehát $-E + i\sqrt{G} = r(\cos\varphi + i\sin\varphi) = U^3 = Ul$. Itt $r = |U^3| = \sqrt{E^2+G}$ és $L = \varphi = \arccos\left(\frac{-E}{r}\right) = \arccos\frac{-E}{\sqrt{E^2+G}}$.

Az U egy értékére a következőket kapjuk: $U = M + i\sqrt[6]{E^2+G}\sin\frac{L}{3}$ ahol $M = \sqrt[6]{E^2+G}\cos\frac{L}{3}$.

Ismeretes, hogy $v = \bar{u}$ és ha $y_1 = (u + v)$, $y_2 = -\frac{1}{2}(u+v) + i\frac{\sqrt{3}}{2}(u - v)$, $y_3 = -\frac{1}{2}(u + v) - i\frac{\sqrt{3}}{2}(u - v)$ kiszámítása közben bevezetjük az $N = \sqrt{3}\sqrt[6]{E^2+G}\sin\frac{L}{3}$ jelölést, valamint figyelembe vesszük az $x = y - \frac{b}{3a}$, helyet-


START
$Ax^3+Bx^2+Cx+D=0$
A, B, C, D beolvasása
A = 0
h
i
B = 0
...
Legfeljebb másodfoku egyenlet l. a 2. ábrát
$F:=\frac{-\left(\frac{B}{A}\right)^2+3\frac{C}{A}}{9}$
$E:=\frac{2\left(\frac{B}{A}\right)^3-9\frac{BC}{A^2}+27\frac{D}{A}}{54}$
$G:=E^2+F^3$
$G\geq 0$
$U1:=-E+\sqrt{G}$
$G:=-G$
$L:=\arccos\frac{-E}{\sqrt{E^2+G}}$
$M:=\sqrt[6]{E^2+G}\cdot\cos\frac{L}{3}$
$N:=\sqrt{3}\,\sqrt[6]{E^2+G}\sin\frac{L}{3}$
$U1\geq 0$
$U:=\sqrt[3]{U1}$
$U:=-\sqrt[3]{-U1}$
$V1:=-E-\sqrt{G}$
$V1\geq 0$
$V:=-\sqrt[3]{-V1}$
$V:=\sqrt[3]{V1}$
$x1:=2M-\frac{B}{3A}$
$x2:=-M-N-\frac{B}{3A}$
$x3:=-M+N-\frac{B}{3A}$
$J:=-\frac{1}{2}(U+V)$
$K:=\frac{\sqrt{3}}{2}(U-V)$
$x1:=-2J-\frac{B}{3A}$
$x2:=J-\frac{B}{3A}+iK$
$x3:=J-\frac{B}{3A}-iK$
STOP

3. ábra

tesítést x 1 = 2 M — b/3 a, x 2 = — M — N — b/3 a, x 3 = — M + N — b/ /3 a valós gyököket kapjuk. A 3. ábrán ezt az esetet is feltüntetve a folyamatábrával elkészültünk. Az ábra alapján bármely valós együtthatójú harmadfokú egyenlet megoldásának lépéseit a megfelelő ágon végigkísérhetjük.

A folyamatábra alapján a legfeljebb harmadfokú egyenletek megoldásának egy lehetséges programja a következő:

```
*comment
  Valoos egyuetthatoos harmadfoku egyenletek megoldaasa;
begin
  real a, b, c, d, v, x, x1, x2, x3, D, D1, D2, D3, E, F, G, U, V, U1, V1, J,
  K, L, M, N;
  format ('⌴⌴⌴111⌴111·111⌴111');
A:
read (a, b, c, d);
if a = 0∧b = 0∧c = 0
  then
  begin
    if d = 0
      then
      begin
        print ('?identikus egyenlet');
        go to A
      end
      else
      begin
        print ('?ellentmondaasos egyenlet');
        go to A
      end
    end;
if a = 0 ∧ b = 0
  then
  begin
    print ('?elsoefoku egyenlet');
    x = — d/c;
    print ('?x =', x);
    go to A
  end;
if a = 0
  then
  begin
    print ('?maasodfoku egyenlet');
D = c↑2 — 4 × b × d;
if D ≥ 0
```

*A kurzívból szedett szavak aláhúzottnak tekintendők.

```
  then
  begin
    x1: = (— c + sqrt (D))/2/b;
    x2: = (— c — sqrt (D))/2/b;
    print ('?x1 =', x1, '?x2 =', x2);
    go to A
  end
  else
  begin
    D1: = — c/2/b;
    D2: = (sqrt (— D))/2/b;
    D3: = —(sqrt (— D))/2/b;
    print ('?x1 =Re', D1, '———Im', D2, '?x2 = Re', D1,'———Im', D3);
    go to A
  end
end;
E: = (2 × b ↑ 3/a ↑ 3 — 9 × b × c/a ↑ 2 + 27 × d/a)/54;
F: = (— b ↑ 2/a ↑ 2 + 3 × c/a)/9;
G: = E ↑ 2 + F ↑ 3;
if G ≥ 0
  then
  begin
    print ('? harmadfoku egyenlet');
    U1: = — E + sqrt (G);
    V1: = — E — sqrt (G);
    if U1 ≥ 0
  then
  begin
    U: = U1 ↑ 0.33333;
    print ('??U = ', U)
  end
  else
  begin
    U: = — (— U1) ↑ 0.33333;
    print ('?U =', U)
  end;
if V1 ≥ 0
  then
  begin
    V: = V1 ↑ 0.33333;
    print ('?V =', V)
  end
  else
  begin
    V: = — (— V1) ↑ 0.33333;
    print ('?V =', V)
  end
J: = — (U +V)/2;
```

```
K: = (U — V) × sqrt (3)/2;
x1: = — 2 × J — b/3/a;
x2: = J — b/3/a;
x3: = — K;
print ('??x1 =⌴⌴', x1, '?x2 = Re', x2,'⌴⌴⌴ Im', K, '?x3 = Re', x2,
'⌴⌴⌴Im', x3);
  go to A
end
else
begin
  G: = — G;
  print ('?Casus irreducibilis');
  L: = arccos (— E/sqrt (E ↑ 2 + G));
  M: = (sqrt (E ↑ 2 + G)) ↑ 0.33333 × cos (L/3);
  N: = sqrt (3) ×(sqrt (E ↑ 2 + G)) ↑ 0.33333 × sin (L/3);
  x1: = 2 × M — b/3/a;
  x2: = — M — N — b/3/a;
  x3: = — M + N — b/3/a;
  print ('?M =', M, '?N =', N, '??x1 =', x1, '?x2 =', x2, '?x3 =', x3);
  go to A
  end
end;
?
```

Gyakorlatokra másod- és harmadfokú egyenleteket jelölünk ki megoldásra — az önálló munka biztosítása érdekében esetleg személyenként különbözőt —, s ezeket órán a program segítségével végrehajtott gépi számolás eredményei alapján ellenőrizzük. Ezt a munkát a tanár számára meggyorsítja, megbízhatóbbá teszi, az esetleges hiba megkeresését is elősegíti, ha a végeredményen (a gyökökön) kívül egyéb közbülső eredményeket is kiíratunk a géppel. Mi, amint a programból is látható, első- és másodfokú egyenleteknél csak a gyököket (utóbbiaknál ha $D < 0$, akkor azok valós és képzetes részét), harmadfokúaknál azonban U és V illetve casus irreducibilis esetén M és N értékét is kiírattuk. Így elképzelhető, hogy a hallgatónak hibás eredmény esetén az adott egyenlet megoldásának csak egy részét kell megismételnie, ugyanis legalább egy harmadfokú egyenlet megoldásának útját a számítások pontosságára is ügyelve, mindenkinek végig kell járnia.

A program alapján $x^3 - 6x^2 + 37x - 196 = 0$ és az $x^3 - 4x + 2 = 0$ (példatárban is szereplő) egyenletek együtthatóinak a beolvasása után a gép rendre a következőket írja ki:

Harmadfokú egyenlet

U = 4.999 920
V = —2.999 967

X1 =	3.999 952	Im	6.928 105
X2 = Re	1.000 024	Im	6.928 105
X3 = Re	1.000 024	Im	—6.928 105

harmadfokú egyenlet, casus irreducibilis

M = .837 564
N = 1.376 752

X1 = 1.675 128
X2 = —2.214 317
X3 = .539 188

Ha az első adott harmadfokú egyenlet megoldása során $U = 5$ és $V = -3$ értékeket a hallgató kiszámította, akkor már csak a két utolsó lépés végrehajtása közben követhetett el hibát, így csak ezeket kell javítania. (Sokszor ilyen esetben a $b/3a$ levonásáról megfeledkeznek.)

A második egyenletnél, ha M és N értéke hibás, akkor már az L vagy a G meghatározása közben is beléphetett a hiba, ilyenkor ezeket kell először ismételt számolással ellenőrizni. (E, F együtthatók, s esetleg még más részeredmény kiíratásával az ellenőrzés precízsége növelhető.)

A gép segítségével végzett ezen számításokat a gyakorlati órán történő feladatmegoldásnál, dolgozatok javításánál, szakkörön egy-egy feladat részfeladataként felmerülő egyenlet megoldásánál stb. is felhasználhatjuk.

A negyedfokú egyenletek hasonló szellemben történő tárgyalása már nem ilyen egyszerű. A bonyolultabb folyamat ábrájának megbeszélése az elméleti ismeretek birtokában a gyakorlati órán történhet. Megoldásukra írt program alapján végzett számítások eredményeit és részeredményeit a gyakorlatvezetők az előzőek szerint használhatják.

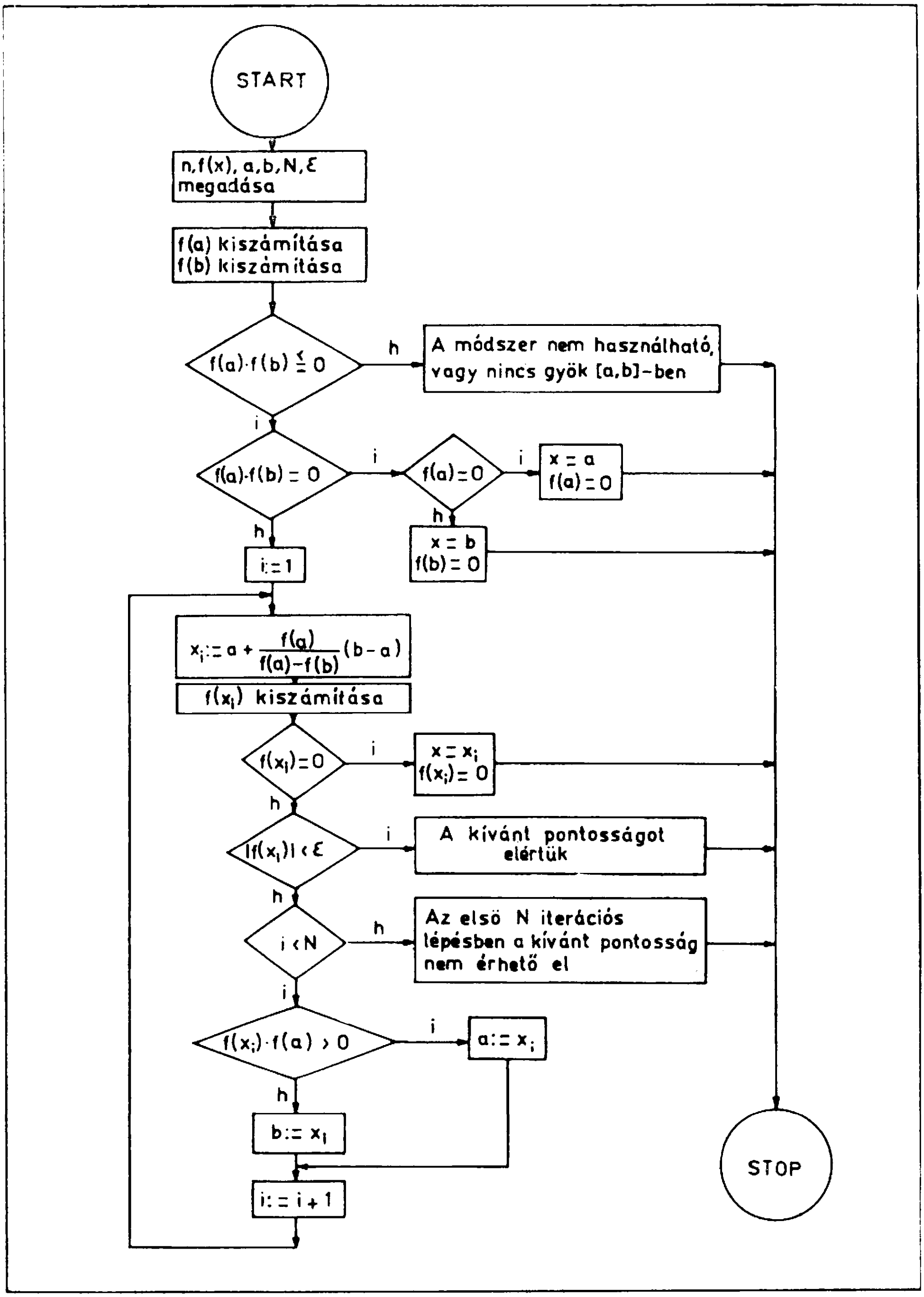
A harmad- és negyedfokú egyenletek megoldása közben a hallgatók is tapasztalhatták, hogy a képletekkel való számolás hosszas és fáradságos, és nem mindig pontos. A gyakorlatban hasznosabbak az általános közelítő módszerek, amelyek akárhányadfokú egyenletre is megadják a kívánt pontosságú megoldást.

Foglalkozzunk a következőkben a valós együtthatós algebrai egyenletek valós gyökeinek közelítő meghatározásával.

Az eddigiek során az algebragyakorlatokon követtük a jegyzetben található felépítést és a gyökök behatárolása után a húr- vagy a Newton-módszerrel kiszámítottuk az egyenlet gyökét előre megadott pontosságra. A módszerek azonban sok numerikus számolást igényelnek és ezért csak néhány feladatot tudtunk megoldani a módszer bemutatására. Eddig az otthoni munka önállóságának ellenőrzése sem volt megnyugtató.

A húr-módszer tanításánál jelenleg a következő utat követjük. Az elméleti bevezetés után a módszer alkalmazását folyamatábrán foglaljuk össze (4. ábra).

A folyamatábra szerint először megadjuk az egyenlet bal oldalán álló polinom fokszámát, az f(x) polinomot és a vizsgált [a, b] intervallumot. Megadjuk azt is, hogy maximálisan hány iterációs lépést végezzünk el (N), továbbá azt, hogy a polinom helyettesítési értéke a közelítő gyökné mennyivel térhet el a zérustól (ε). Ezután megvizsgáljuk az f(a) és f(b) helyettesítési értékeket. Ha egyenlő előjelűek, akkor a módszer nem



START
n, f(x), a, b, N, ε megadása
f(a) kiszámítása
f(b) kiszámítása
f(a)·f(b) ≤ 0
h
A módszer nem használható, vagy nincs gyök [a,b]-ben
i
f(a)·f(b) = 0
i
f(a) = 0
i
x = a
f(a) = 0
h
x = b
f(b) = 0
h
i := 1
x_i := a + f(a)/(f(a)−f(b)) (b − a)
f(x_i) kiszámítása
f(x_i) = 0
i
x = x_i
f(x_i) = 0
h
|f(x_i)| < ε
i
A kívánt pontosságot elértük
h
i < N
h
Az elsö N iterációs lépésben a kívánt pontosság nem érhető el
i
f(x_i)·f(a) > 0
i
a := x_i
h
b := x_i
i := i + 1
STOP

4. ábra

használható (lásd az alábbi példát), ha pedig az egyik zérus, megtaláltuk az egyenlet adott intervallumba eső gyökét.

Ha $f(a) \cdot f(b) < 0$, akkor kiszámítjuk az első közelítő gyököt (x_i-t, ahol jelen esetben $i = 1$) az ismert $x_i = a + \frac{f(a) \cdot (b - a)}{f(a) - f(b)}$ összefüggés alapján és megvizsgáljuk az $f(x_i)$ helyettesítési értéket. Ha még nem értük el a kívánt pontosságot, akkor i értékét eggyel növeljük és megismételjük az eljárást azzal a változtatással, hogy x_i tölti be „a" vagy „b" szerepét, aszerint, hogy $f(x_i)$ előjele f(a) vagy f(b) előjelével egyezik meg. Az eljárást addig folytatjuk, míg vagy elérjük a kívánt pontosságot, vagy pedig i eléri a megengedett N értékét.

A folyamatábra alapján a következő ALGOL 1204-es programot készítettük el:

```
comment
  Polinom gyoekenek meghataarozasa adott intervallumban hur-mod-
  szerrel;
D:
begin
  integer n, N;
  real a, b, p, q, r;
  read (n);
  begin
    real x, epsz;
    integer i;
    array A [0:n];
    real procedure f(x, n, A);
      real x;
      integer n;
      array A;
      f: = if n = 0 then A [0] else f(x, n — 1, A) × x + A [n];
    read (A, a, b, N, epsz);
    format ('12');
    print ('??Fokszaam: ',n. 'az egyenlet bal oldala:?');
    for i: = 0 step 1 until n do
      begin
        format ('+ 1.234₁₀1');
        print (A[i]);
        format ('1');
        print ('x ↑ ', n — i, ' ')
      end;
    format ('1.12₁₀1');
    print ('?Az intervallum: [',a,' ; ',b,'].');
    format ('?x12 = —1.2345₁₀12¯¯¯¯f(x12) = 1.234¯567₁₀12');
    p: = f(a, n, A);
    q: = f(b, n, A);
    if p × q > 0
```

```
        then go to B;
      if p × q = 0
        then go to C;
      i: = 1;
    L:x: = a + (p/(p — q)) × (b — a);
      r: = f(x, n, A);
      print (i, x, i, r);
      if abs (r) < epsz
        then go to H;
      if i ≥ N
        then go to K;
      if r × p > 0
        then
        begin
          a: = x;
          p: = r
        end
        else
        begin
          b: = x;
          q: = r
        end;
      i: = i + 1;
      go to L
    end;
  B:
    print ('? A moodszer nem hasznaalhatoo, vagy nincs valoosgyoek az
    [a. b] intervallumban');
    go to P;
  C:
    if p = 0
      then print ('?x =' ,a, 'f(x) = 0')
      else print ('?x = ',b,' f(x) = 0');
    go to P;
  H:
    print ('? A kivaant pontossaagot eleertuek');
    go to P;
  K:
    print ('? Az elsoe N iteraacioos leepeessel a kivaant pontossaag nem
    eerhetoe el');
  P:
    print ('???');
    go to D
  end;
  ?
```

Konkrét feladat megoldása esetén az adatszalagon a következő számokat kell megadni a következő sorrendben. n (az egyenlet fokszáma), A (a nullára redukált egyenlet bal oldalán álló polinom $n+1$ együtthatója, az n-edfokú tag együtthatójával kezdve), a, b (a vizsgált intervallum alsó illetve felső végpontja), N (az i-re megadott felső határ), ε (a közelítő gyök helyettesítési értékétől kívánt pontosság).

A program kiírja a megadott egyenlet fokszámát, az egyenlet bal oldalát és a vizsgált intervallumot, majd sorba kiírja az x_i közelítő gyököket és ezek helyettesítési értékeit. Megállásnál mindig közli a megállás okát („A kívánt pontosságot elértük", vagy „A módszer nem használható" stb.). Például az $x^3 - 3x^2 + 4 = 0$ egyenlet paramétereinek a bevitele után a gép a következőket írja ki ($N = 6$ és $\varepsilon = 10^{-4}$):

Fokszám: 3 az egyenlet bal oldala

$+1.000\ 0x \uparrow 3 - 3.000\ 0x \uparrow 2 + \ .000\ 0x \uparrow 1 + 4.000\ 0x \uparrow 0$

Az intervallum: $[1.50_{10}0;\ 2.50_{10}0]$.

A módszer nem használható, vagy nincs valós gyök az intervallumban.

Az egyenletnek ugyanis az intervallumban egy kétszeres gyöke van ($x = 2$), amit húr-módszerrel nem tudunk meghatározni. Az $x^4 + 3x^3 + 4x^2 + x - 3 = 0$ egyenletre a program a következő közelítő értékeket sorolja fel ($a = 0$, $b = 1$, $N = 6$, $\varepsilon = 10^{-4}$):

$x1 = 3.3333 \times_{10}-1$	$f(x1) = -2.098_{10}0$
$x2 = 5.0610_{10}-1$	$f(x2) = -1.014_{10}0$
$x3 = 5.7755_{10}-1$	$f(x3) = -3.989_{10}-1$
$x4 = 6.0389_{10}-1$	$f(x4) = -1.436_{10}-1$
$x5 = 6.1316_{10}-1$	$f(x5) = -5.009_{10}-2$
$x6 = 6.1636_{10}-1$	$f(x6) = -1.726_{10}-2$

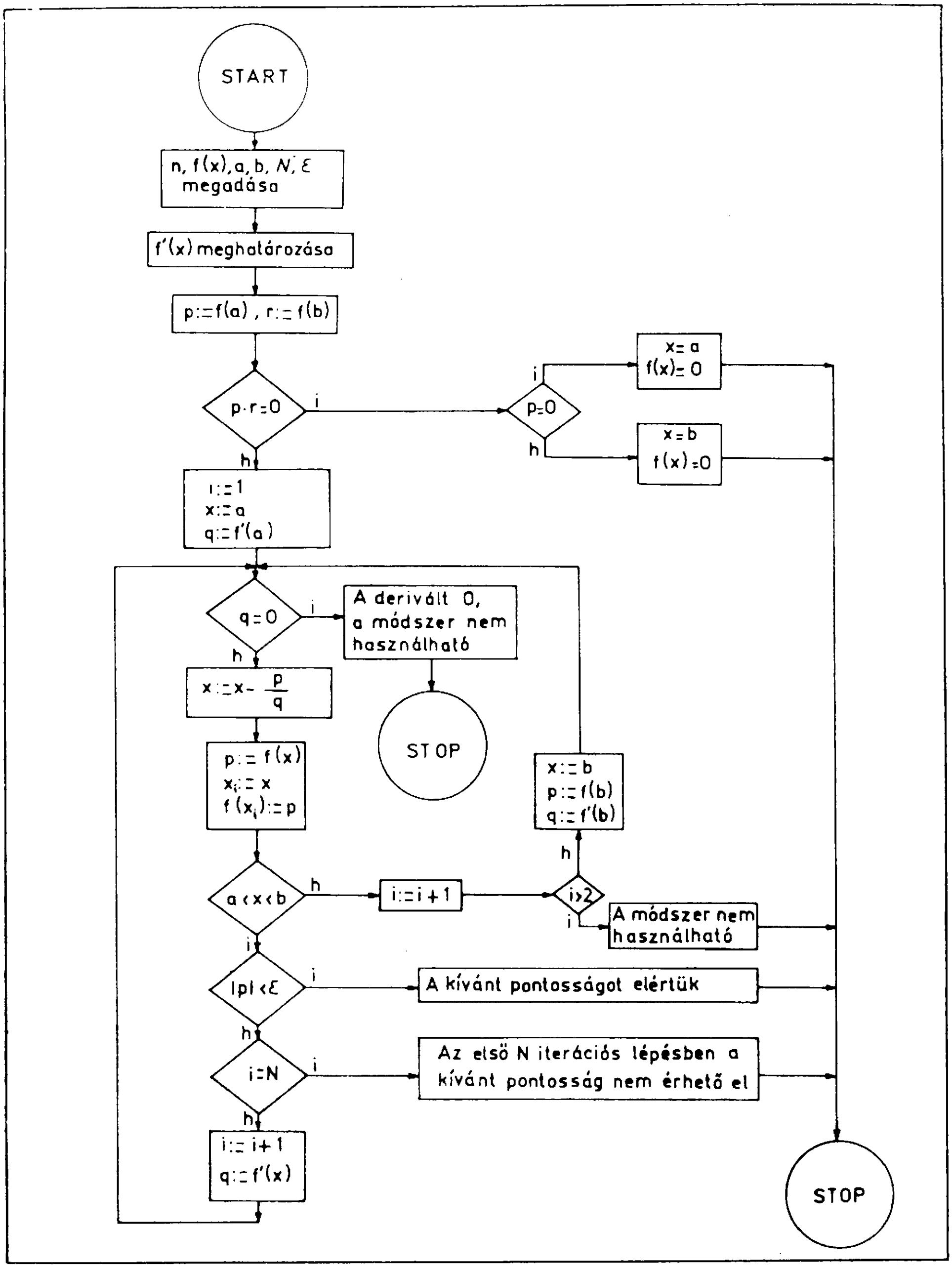
Az első N iterációs lépéssel a kívánt pontosság nem érhető el.

Ezt a programot a gyakorlatokon jól fel tudjuk használni, különösen az otthoni munka ellenőrzésére. Minden hallgatónak különböző feladatot adunk, melyek ha másban nem, a megadott intervallumban különböznek, így mindenki kénytelen önállóan dolgozni. Ellenőrzéskor a géppel kiszámíttatott eredmények alapján elég mindenkitől csak egy részletet, vagy valamelyik közelítő gyököt számon kérni, így is kiderül, hogy jól számolt-e az illető hallgató.

Az előzőekhez hasonlóan dolgoztuk fel a Newton-módszert is. Ennek folyamatábrája az 5. ábrán látható.

Az ALGOL 1204-es programja a következő:

```
comment
  Egyenlet gyoekeinek meghataarozaasa adott intervallumban Newton-
  modszerrel;
```


START
n, f(x), a, b, N, ε megadása
f'(x) meghatározása
p:=f(a), r:=f(b)
p·r=0
i
h
p=0
x=a f(x)=0
x=b f(x)=0
i:=1 x:=a q:=f'(a)
q=0
A derivált 0, a módszer nem használható
STOP
x:=x− p/q
p:=f(x) x_i:=x f(x_i):=p
a<x<b
i:=i+1
i>2
x:=b p:=f(b) q:=f'(b)
A módszer nem használható
|p|<ε
A kívánt pontosságot elértük
i=N
Az első N iterációs lépésben a kívánt pontosság nem érhető el
i:=i+1 q:=f'(x)
STOP

5. ábra

```
L:
begin
  integer i, n, N, k;
  real a, b;
  read (n);
  begin
    real x, p, q, r, epsz;
    array A, D [0:n];
real procedure f(x, n, A);
  real x;
  integer n;
  array A;
  f: = if n = 0 then A[0] else f(x, n — 1, A) × x + A[n];
read (A, a, b, N, epsz);
format ('12');
print ('??Fokszaam:' ,n, 'a polinom:?');
for i: = 0 step 1 until n do
  begin
    format ('+ 1.234₁₀1'):
    print (A[i]);
    format ('1');
    print ('x ↑ ' ,n — i, ' ')
  end;
format ('1.12₁₀1');
print ('?Az intervallum: [',a,' ; ',b,'].');
format ('?x12 = —1.2345₁₀12————f(x12) = 1.234₁₀12');
D[0]: = 0;
for k: = 0 step 1 until n — 1 do
  D[k + 1]: = A[k] × (n — k);
p: = f(a, n, A);
r: = f(b, n, A);
q: = f(a, n, D);
if p × r = 0
  then
  begin
    if p = 0
      then go to E
      else go to F
    end;
  i: = 1;
  x: = a;
B: if q = 0
    then
    begin
      print ('?A derivalt 0');
      go to H
    end;
  x: = x — p q;
```

```
  p: = f(x, n, A);
  print (i, x, i, p);
  if a < x ∧ x < b
    then go to G;
  i: = i + 1;
  if i > 2
    then go to H;
  x: = b;
  p: = r;
  q: = f(b, n, D);
  go to B;
G: if abs (p) < epsz
    then go to I;
  i: = i + 1;
  if i > N
    then go to J;
    q: = f(x, n, D);
    go to B
  end;
E:
  print ('?x =' ,a, 'f(a) = 0');
  go to K;
F:
  print ('?x =' ,b, 'f(b) = 0');
  go to K;
H:
  print ('?A modszer nem hasznaalhato, vagy nincs gyoek az interval-
  lumban');
  go to K;
I:
  print ('?A kivaant pontossaagot eleertuek');
  go to K;
J:
  print ('?Az elsoe' ,N, 'iteraacios leepeesben a kivaant pontossaag nem
  eerhetoe el');
K:
  print ('???');
  go to L
end;
?
```

Működése hasonló a húr-módszer programjának működéséhez. Az egyenletnek itt is ugyanazokat a paramétereit kell megadni és ugyanabban a sorrendben. A kiírás formátuma is hasonló. Itt a gép akkor írja ki, hogy „A módszer nem használható, vagy nincs gyök az intervallumban", ha valamelyik x_i közelítő gyök $i \geq 2$ esetén nincs a megadott intervallumban, vagy $f'(x_i) = 0$.

A program szerint a gép az intervallum baloldali végpontját használja fel x_i meghatározására, ezért az otthoni munka kiadásánál fel kell hívni a hallgatók figyelmét arra, hogy ők így kezdjék, mert ellenkező esetben a gép és a hallgatók által kiszámolt x_i értékek különböznek és csak a végeredményt tudjuk ellenőrizni.

A húr-módszer tárgyalásakor bemutatott egyenleteket a Newton-módszer programjával megoldva a következőket írja ki a gép. (A paraméterek ugyanazok, a gép által kiírt polinomtól és intervallumtól eltekintünk):

Az $x^3 - 3x^2 + 4 = 0$ egyenlet esetén:

$x1 = 1.7778_{10}00$	$f(x1) = 1.371_{10}{-01}$
$x2 = 1.8935_{10}00$	$f(x2) = 3.280_{10}{-02}$
$x3 = 1.9478_{10}00$	$f(x3) = 8.045_{10}{-03}$
$x4 = 1.9741_{10}00$	$f(x4) = 1.993_{10}{-03}$
$x5 = 1.9871_{10}00$	$f(x5) = 4.961_{10}{-04}$
$x6 = 1.9936_{10}00$	$f(x6) = 1.237_{10}{-04}$

Az elso 6 iteracios lepessel kivant pontossag nem erheto el.

Az $x^4 + 3x^3 + 4x^2 + x - 3 = 0$ egyenlet esetén pedig:

$x1 = 3.0000_{10}00$	$f(x1) = 1.980_{10}+02$
$x2 = 7.2727_{10}{-01}$	$f(x2) = 1.276_{10}00$
$x3 = 6.2994_{10}{-01}$	$f(x3) = 1.246_{10}{-01}$
$x4 = 6.1819_{10}{-01}$	$f(x4) = 1.657_{10}{-03}$
$x5 = 6.1803_{10}{-01}$	$f(x5) = 3.036_{10}{-07}$

A kivant pontossagot elertuk.

Ezek a kiírási formátumok az előzőekhez hasonlóan sok segítséget adnak a gyakorlatvezető tanároknak a hallgatók munkájának ellenőrzéséhez.

FELHASZNÁLT IRODALOM

Dr. Szendrei János: Algebra és számelmélet. Tankönyvkiadó, Bp. 1970.

Lőcs Gyula: ALGOL 60. Műszaki Könyvkiadó, Bp. 1971.

Kiss Ottó—Vértessy Péter: Numerikus és grafikus módszerek, Tankönyvkiadó, Budapest, 1966.

EINE ART DER BEHANDLUNG VON GLEICHUNGEN HÖHEREN GRADES MIT BENÜTZUNG VON ELEMENTEN DER RECHNUNGSTECHNIK

Kiss Péter und Szepessy Bálint

Die Verfasser demonstrieren in der Arbeit eine Behandlungsart eines Kapitels des Algebramaterials der Hochschule mit Benützung von Elementen der Rechnungstechnik. Sie beschreiben die Möglichkeit zur Vertiefung des theoretischen Materials mit Anwendung von Prozessfiguren. Sie zeigen einige ALGOL-Programme vor, die hauptsächlich vom übungführenden Lehrer zur Kontrolle der Arbeit der Studenten verwendet sein können.