

нее число аппаратов, занятых или свободных от обслуживания и т.д. Однако наиболее целесообразно использовать экономические показатели, которые дают обобщенную характеристику производственных процессов.

Иногда при планировании и организации производственных процессов необходимо учитывать значительно больше случайных факторов. В этом случае хорошие результаты могут быть получены с помощью методов факторного анализа.

#### Список литературы

1. Гольштейн Е.Г., Юдин Д.Б. Задачи линейного программирования транспортного типа. — М.: Наука, 1969. — 256 с.
2. Кожин А.П. Математические методы в планировании и управлении грузовыми перевозками. — М.: Высш. шк., 1991. — 296 с.

ЭОЖ 510.67

А.Е.Сланбекова<sup>1</sup>, Ш.К.Каменова<sup>2</sup>

<sup>1</sup>Е.А.Бекетов атындағы Қарағанды мемлекеттік университеті;

<sup>2</sup>№ 4 орта мектебі, Қарағанды

#### АЛГЕБРАЛЫҚ СЫЗЫҚТЫҚ ЕМЕС ТЕНДЕУЛЕРДІ ШЕШУДЕ ҚОЛДАНЫЛАТЫН ПРОГРАММАЛАУ ТІЛДЕРІН ТАЛДАУ

*Рассмотрены численные методы решения нелинейных уравнений, приведены программы вычисления корней нелинейных уравнений на языке Turbo Pascal и Фортран, дан сравнительный анализ изученных методов и программ.*

*Present work explains numerical methods of solving nonlinear equations. Contains Turbo Pascal and Fortran programs for calculation solutions of nonlinear equations and benchmark analysis of these methods and programs.*

Осы жұмыстың мақсаты — алгебралық және трансценденттік теңдеулерді жуықтап шешу әдістерін табу. Бұл әдістермен шешудің Turbo Pascal және Фортран тілдерінде бағдарламасын жасап, есептеулерді құрып, оларды салыстырып талдау.

Жұмыстың негізгі түсініктерін қарастырайық.

$$f(x) = 0 \quad (1)$$

теңдеуі берілген болсын, мұндағы  $f(x)$  функциясы кейбір  $a < x < b$  аралығында анықталған және үздіксіз.  $x = \xi$  әрбір мағынасында (1) теңдеудің түбірі немесе  $f(x)$  функциясының нөлі деп аталады.

Егер  $f(x)$  функциясы көпмүшелік болса, онда (1) теңдеуді алгебралық деп атайды, ал  $f(x)$  функциясын тригонометриялық, логарифмдік т.б. функциялар енетін болса, онда (1) теңдеуді трансценденттік деп атайды [1].

Функцияны нольге айналдыратын кез келген  $x$  мәнін (1) теңдеудің түбірі деп атайды [1]. (1) теңдеудің дәл түбірін дербес жағдайда ғана табуға болады. Сондықтан (1) түрдегі теңдеуді сандық шешу тәсілдерімен жасауға болады. Ол тәсіл көмегімен (1) теңдеудің шешімін жуықтап табу мүмкіндігі туады.

Бұл жағдайда екі есепті шешуге тура келеді [2]:

- 1) түбірді оқшаулау, яғни теңдеудің бір ғана түбірін қамтитын жеткілікті түрде кішкене интервалды анықтау;
- 2) түбірді берілген дәлдікпен анықтау.

(1) теңдеудің нақты түбірі жататын облысты анықтауда егер белгілі бір кесіндінің екі шеткі нүктелерінде әртаңбалы мәндер қабылдайтын болса, онда бұл кесіндіде  $f(x) = 0$  функциясының ең болмағанда бір түбірі болатын қасиетіне сүйенеміз.

1)  $f(x) = 0$  теңдеудің бір ғана түбірі жататын облысты анықтау үшін, масалы, графикалық тәсілді пайдалануға болады.

2) Екінші есепті шешудің көптеген тәсілдері бар, біз келесі тәсілдеріне тоқталамыз [1]:

– қақ бөлу әдісі;

– хорда әдісі;

– ньютон әдісі.

Бұл тәсілдерде бір ғана түбір жататын  $[a, b]$  кесіндісі белгілі деп болжаймыз. Осы түбірді табу үшін Turbo Pascal және Фортран тілдерінде бағдарламасын жасаймыз.

Компьютерде есептерді шығаруда төмендегідей кезеңдерге бөлуге болады:

1. Есептің қойылуы. Оны тұтынушы өзі тұжырымдайды немесе тапсырма түрінде алады.

2. Есептің математикалық сипатталуы.

3. Есепті шешудің алгоритмін жасау.

4. Белгілі бір алгоритмдік тілде программа жасау.

5. Бастапқы деректерді даярлау.

6. Программа мен бастапқы деректерді енгізу.

7. Программа қателерін түзету.

8. Компьютерде есепті шығару және нәтижелерін өңдеу.

Біздің қарастырылған жағдайларымызда есептің математикалық тұжырымдалуы белгілі болады, сондықтан 1 және 2 кезеңдерде орындау қажеттігі жоқ, бірден есепті шешудің алгоритмін жасауға кірісеміз. Алгоритм деп айнымалы шамалардың сандық мәндеріне арифметикалық, логикалық амалдардың тізбегін қолдану арқылы есептің шешіміне бастапқы деректердің жеткілікті диапазонда өзгергендігі нәтижеге жеткізетін амалдар жиынтығын айтамыз. Сонымен, алгоритмді жасаған кезде математикалық тұжырымдау есепті шығару процедурасына түрленеді, ол арифметикалық амалдар мен олардың арасындағы логикалық байланыстар тізбегінен тұрады.

Паскаль тілін 1968–1971 жылдары швейцариялық ғалым Никлаус Вирт оқып-үйренуге қолайлы программалау тілі ретінде ұсынған болатын. Бұл тілдің стандарты кейінірек бекітілді, ол сол кездерде кең таралған АЛГОЛ, ФОРТРАН, БЕЙСИК тілдеріне қарағанда жетілдірілген, жұмыс істеуге ыңғайлы тіл болды. Паскаль тілі өзінің қарапайымдылығының және тиімділігінің арасында дүние жүзіне тез таралды. Қазіргі кезде барлық дербес компьютерлер осы тілде жұмыс істей алады. Паскаль тілінде жазылған программаның дұрыстығын компьютерде тексеру және жіберілген қатені түзету оңай.

Бұл тілде жазылған программа компьютерде орындалу барысында алдымен трансляцияланады, объектік программаға түрлендіріледі де, содан кейін ғана орындалады. Осы сәтте компьютерде программаның екі нұсқасы болады, оның біріншісі алгоритмдік тілдегі алғашқы түпнұсқасы, ал екіншісі — объектік кодтағы жазылған программа. Есеп нәтижесін машиналық кодта жазылған программа арқылы аламыз, ал программаны түзету қажет болғанда, оның алгоритмдік тілде жазылған алғашқы нұсқасы өңделеді.

Қазіргі кезде Паскаль тілі кез келген күрделі есептерді шығара алатын, кең таралған стандартты оқу тіліне айналды. Сондықтан жалпы білім беретін мектептерде программалауды оқытуда осы Паскаль тілі таңдалып алынған. Енді осы тілдің ерекшеліктері мен мүмкіндіктеріне тоқталып өтейік.

Программалар белгілі бір мәселені, есепті шешуге арналған. Есеп шығару барысында компьютерге бастапқы мәліметтер енгізіледі, олар қалай өңделетіндігі көрсетіледі және нәтиже қандай түрде, қандай құралға шығарылатыны туралы айтылады.

Паскаль тіліндегі программа жеке-жеке жолдардан тұрады. Олардың теру, түзуге арнайы мәтіндік редакторлар арқылы атқарылады. Программа алдындағы азат жол немесе бос орын саны өз қалауымызша алынады. Бір қатарға бірнеше команда немесе оператор орналса алады, олар бір-бірімен нүктелі үтір арқылы ажыратылып жазылады, бірақ бір жолда бір ғана оператор тұрғаны дұрыс, ол түзуге жеңіл, әрі оқуға ыңғайлы.

Фортран тілі процедура бағытталған, ол көптеген инженерлік және ғылыми-техникалық мазмұндағы есептерді шешу үшін қолданылады. Бұл тілді сипаттайтын алғашқы мәлімдемені 1954 жылы профессор Дж.Бэкустың басшылығымен бір топ американ мамандары жасады. Фортран (FORTRAN) алғашқы FORMULA TRANSLATOR деген сөздерінің алғашқы буындарынан құрастырылып алынған, «формула аударғыш» деген мағынаны береді. Тілдің тез, кеңінен тарауына бәрінен бұрын оның қарапайымдылығы, конструкцияларының жазылуы кәдімгі математикалық жазбаға ұқсайтындығы, сонымен қатар мәліметтерді енгізу-шығарудағы үлкен мүмкіншіліктер, ғылыми ішкі программалар библиотекасының кеңдігі жатады.

Фортран тілінде программалау курсы университеттің математика, физика және химия факультеттерінің студенттеріне оқытылады. Фортран тілінің нәтижелігі мен дәлдігінің басқа тілдермен салыстырғанда жоғары деңгейде болуына байланысты, студенттер курстық, дипломдық жұмыстарды Фортран тілінде программалауды жөн көреді.

Кейінгі жылдарда тілдің үздіксіз дамуына компьютердің сыртқы пішіні ғана емес, мазмұны мен функциялары өзгеріп, программалық жабдықтар мен операциялық жүйелердің өзгеруі көп себеп болды. Мысалы, Фортран-90 стандартты бұрынғы пайдаланған Фортран-70 стандартынан әлдеқайда өзгерген.

Қақ бөлу әдісінде [2]  $f(x) = 0$  теңдеуі берілсін, мұндағы  $f(x)$  функциясы  $[a, b]$  кесіндісінде үздіксіз және  $f(a) \cdot f(b) < 0$ .

Егер  $f(a)$  функциясының түрі жеткілікті түрде күрделі болса, онда Ньютон әдісінде есептеуде қолданылатын  $f'(x)$  және хорда әдісте жинақтылығын шамалайтын  $\phi'(x)$  туындыларын табу қиындап кетеді. Мұндай жағдайда қақ бөлу әдісін қолдануға болады. Есептеуді көп қажет еткенмен, іздеп отырған нәтижеге алып келеді.

(1) теңдеудің  $[a, b]$  кесіндісінде жатқан шешімін табу үшін кесіндіні тең бөлеміз, яғни бастапқы жуықтау үшін  $x_0 = (a + b) / 2$  тең етіп аламыз. Егер  $f(x) \neq 0$  болса, онда  $[a, x_0]$ ,  $[x_0, b]$  кесіндісінің қайсысының шеткі нүктелерінде функция әр түрлі таңба қабылдаса, сонысын таңдаймыз. Таңдап алған кесіндіні тағыда екіге бөлеміз.

Кесіндіні тең бөлу процесін функцияның мәндері әр түрлі таңбалы болып келетін кесіндінің ұзындығы алдын ала берілген  $\varepsilon$  санынан кіші болғанша жалғастырамыз.

Қақ бөлу әдісін программа түрінде өрнектеуге тырысайық. Программа үшін оны қолдану нұсқауы болуы тиіс.

Ньютон әдісінде [2]  $f(x) = 0$  теңдеудің  $[a, b]$  кесіндісінде бір ғана түбірі болсын, сонымен қатар  $f'(x)$  және  $f''(x)$  анықталған, үздіксіз және  $[a, b]$  кесіндісінде таңбасы тұрақты дейік.

Ньютон әдісін қолданып,  $f(x) = x^3 + 4x^2 - 3x - 5 = 0$  теңдеуінің  $[1, 2]$  кесіндісінде жуық түбірін  $\varepsilon = 0,001$  дәлдігімен есептейтін программа құрып, нәтижені экранға шығарыңыз.

$$f'(x) = 3x^2 + 8x - 3.$$

Жуықтап есептеу формуласын төмендегі түрде анықтаймыз:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}.$$

Бізде мысал үшін

$$x_n = x_{n-1} - \frac{x^3 + 4x^2 - 3x - 5}{3x^2 + 8x},$$

мұндағы  $n = 1, 2, \dots$

Бастапқы  $x_0$  жуықтаудан  $f(x_0)f''(x_0) > 0$  шарты орындалатындай есепті анықтаймыз, себебі

$$f''(x) = 12x + 8 > 0 \quad \forall x \in [1, 2],$$

ал енді  $x_0$  нүктесін таңдай білуіміз керек.

Есепті  $|x_n - x_{n-1}| < 10^{-2}$  дәлдікке жеткенге дейін тексереміз.

Есептегі  $n$  санының мәні орындалған итерациялар санына тең болады.

Хорда әдісінде [2] де осылай алгоритмін құрғанда кезеңдерге бөліп жазамыз. Қысқаша теориялық мағлұматтан соң, бірден мысалды шешуге кірісеміз. Сонымен, кез келген лабораториялық жұмыстарды жоғарыда айтылған кезеңдерге бөліп ашып жазуға болады. Қысқаша сөздік алгоритмде не істейтініңді анық түсініп алған соң, оның негізінде сүйене отырып, тілдің ережесі мен операторлар негізінде программа құрамыз.

Компьютерді игеру екінші сауаттылықты игеру болып табылады. Ал алгоритмдік тіл арқылы оның көптеген қызық қырларын ашамыз және басқа мүмкіндіктерін оқып-үйренгенде, өзінді еркін-рек сезінетін боласың.

Сонымен, сонында айтарымыз Паскаль тілі программалауды енді үйреніп жатқандарға қолайлысы болып табылады. Сол себепті да бұл тілдегі программаларда қолданылатын айнымалылар типін алдын ала сипаттау міндетінің талабы расталған.

Ал Фортран тілі қолданбалы есептеулермен айналысатын тәжірибелі мамандарға арналған. Бұған дәлел кейбір айнымалыларды сипаттау міндетті емес — бұл жағдайда үнсіз келісім ережесі қолданылады. Фортран тіліндегі біздің программаларда үнсіз келісім ережесі бойынша  $n$  айнымалысын пайдаланамыз. Үнсіз келісім ережесін қолдану тәжірибелі маман уақытының бір уақыттың бөлігін босатады және сол уақытта есепті шешу алгоритміне қатысты программада жұмыс істеуге мүмкіндік береді.

Есептің Pascal тілінде орындалуы [3].

$3x^3 + 4x^2 - 12x - 5 = 0$  теңдеуін 0,01 дәлдікпен қарқ бөлу әдісі бойынша шешу программасы былай жазылады.

```
uses crt;
const e=0.01;
label 1,2;
var a,b,c:real;
function f(x:real):real;
begin
f:=3*x*x*x+4*x*x*x-12*x*x-5;
end;
begin
clrscr;
a:=1;
b:=2;
if f(a)*f(b)>0 then begin
write('tubiri jok');
goto 2; end;
1: c:=(a+b)/2;
if f(a)*f(c)>0 then
a:=c else b:=c;
if (abs(a-b)>e) then goto 1;
write('a=',c);
2:end.
```

Жауабы:  $a=1,5928$   
7-қадамда орындалды.

$x^3 + 0,1x^2 + 0,4x - 1,5$  теңдеуін 0,001 дәлдікпен хорда әдісі бойынша шешу программасы былай жазылады.

```
uses crt;
const eps=0.001;
label 1,2,3;
function f1(x:real):real;
begin
clrscr;
f1:=x*x*x-0.1*x*x+0.4*x-1.5;
end;
function f2(x:real):real;
begin
clrscr;
f2:=3*x*x-0.2*x+0.4;
end;
var x,x1,x2,a,b:real;
begin
a:=1;
b:=2;
if (f1(a)*f2(a)>0) then goto 1;
x1:=a;
```

```

2: x:=x1-(f1(x1)*(b-f1(x1))/(f1(b)-f1(x1)));
if abs(x-x1)>eps then begin
  x1:=x;
goto 2; end; goto 3;
1: x1:=b;
  x:=x1-(f1(x1)*(x1-a))/(f1(x1)-f1(a));
  if (abs(x-x1)>eps) then begin x1:=x;
    goto 3;
  end;
3: writeln('x=',x:6:3);
  readkey;
end.

```

Жауабы:  $x=1.059$   
6-қадамда орындалды.

$2x^3 + 4x^2 - 3x - 5 = 0$  теңдеуін 0,001 дәлдікпен ньютон әдісі бойынша шешу программасы былай жазылады.

```

uses crt;
const eps=0.001;
label 1;
function f(x:real):real;
begin
  clrscr;
  f:=2*x*x*x+4*x*x-3*x-5;
end;

function f1(x:real):real;
begin
  clrscr;
  f1:=3*x*x+4*x-3;
end;

function f2(x:real):real;
begin
  clrscr;
  f2:=12*x+8;
end;

var x,x1,x2,a,b:real;
begin
  a:=1;
  b:=2;
if (f(a)*f2(a))>0 then x1:=a else x1:=b;
1: x:=x1-(f(x1)/f1(x1));
  if (f(a)<>0) then begin
    if abs(x-x1)>eps then begin
      x1:=x;
      goto 1; end; end
    else x:=a;
  writeln('x=',x:7:5);

  readkey;
end.

```

Жауабы:  $x=1.15831$   
5-қадамда орындалды.

Есептің Фортран тілінде орындалуы [4].

$3x^3 + 4x^2 - 12x - 5 = 0$  теңдеуін 0,01 дәлдікпен қаж бөлу әдісі бойынша шешу программасы былай жазылады.

```

program pop
real*8, x,a,b,c,eps/1e-2/
f(x)=3*x**4+4*x**3-12*x**2-5
print *,'a,b?'
read *,a,b
n=0
c f(x)=f
c f=f(x)
if (f(a)*f(b).gt.0) then
  print *,'tybiri jok'
  stop 1
end if
1 n=n+1
  c=(a+b)/2
  if (f(a)*f(c).gt.0) then
    a=c
  else
    b=c
  end if
  if (abs(a-b).gt.eps) goto 1
print *,'c=',c, ' n=',n
stop 0
end

```

Жауабы: a,b?  
c = 1.5859375000000000      n = 7  
Return code 0

$x^3 + 0,1x^2 + 0,4x - 1,5$  теңдеуін 0,001 дәлдікпен хорда әдісі бойынша шешу программасы былай жазылады:

```

program horda
real*8, x,x1,f1,f2,a,b,eps/1e-3/
f1(x)=x**3-0.1*x**2+0.4*x-1.5
f2(x)=3*x**2-0.2*x+0.4
c print *,'a,b?'
c read *,a,b
n=0
a=1
b=2
c f1(x)=f1
c f1=f1(x)
if (f1(a)*f2(x).gt.0) goto 1
x1=a
2 n=n+1
x=x1-(f1(x1)*(b-f1(x1)))/(f1(b)-f1(x1))
if (abs(x-x1).gt.eps) then
  x1=x
  goto 2
end if

```

```

goto 3
1 x1=b
x=x1-(f1(x1)*(x1-a))/(f1(x1)-f1(a))
if (abs(x-x1).gt.eps) then
  x1=x
  goto 3
end if
3 print *,'x=',x,' n=',n
stop 0
end

```

Жауабы:

x = 1.059255587055197      n = 6

Return code 0

$2x^3 + 4x^2 - 3x - 5 = 0$  теңдеуін 0,001 дәлдікпен ньютон әдісі бойынша шешу программасы келесі түрде жазылады.

```

program nuton
real*8, x,a,b,f,f1,f2,eps/1e-3/
f(x)=2*x**3+4*x**2-3*x-5
f1(x)=6*x**2+8*x-3
f2(x)=12*x+8
print *,'a,b?'
read *,a,b
n=0
c f(x)=x
c x=f(x)
if (f(a)*f2(a).gt.0) then
  x1=a
else
  x1=b
end if
1 x=x1-f(x1)/f1(x1)
n = n + 1
if (f(a).ne.0) then
  if (abs(x-x1).gt.eps) then
    x1=x
    goto 1
  end if
else
  x = a
end if
print *,'x=',x,' n=',n
stop 0
end

```

Жауабы: a,b?

x = 1.158312395179365      n= 5

Return code 0

Бұл жұмыстың маңыздылығы, жалпы білім беретін жоғары оқу орындарында өтілетін пәндердің ішінде математика мен физика ерекше орын алады, себебі бұл пәндер ғылым мен техниканың қай саласымен болса да біте қайнасып жатыр. Сондықтан жоғары оқу орын бітірушілер теориялық мәселелермен қатар, оларды практикада да қолдана білуі өте қажет және арнайы осы саланың мамандарына оқу кезінде қолдануына болады.

## Әдебиеттер тізімі

1. Пантелеев А.В., Киреев А.В. Численные методы в примерах и задачах. — М.: Высш. шк., 2004. — 480 с.
2. Демидович Б.П., Марон И.А. Основы вычислительной математики. — М.: Наука, 1966. — 784 с.
3. Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. — 2-е изд., перераб. и доп. — СПб.: БХВ–Санкт-Петербург, 1999. — 416 с.
4. Бартедьев О.В. Фортран для студентов. — М.: Диалог-МИФИ, 1999. — 400 с.

УДК 510.53

Д.А.Тусупов

Таразский государственный университет им. М.Х.Дулати

**ОПРЕДЕЛИМЫЕ ОТНОШЕНИЯ, ИЗОМОРФИЗМЫ И СЕМЕЙСТВА СКОТТА  
СТРУКТУРЫ С ДВУМЯ БИНАРНЫМИ ПРЕДИКАТАМИ**

*Симметриялық, иррефлексивтік графты екі орындық екі предикатты құрылымға анықтама-лық әдіс жасалған. Осы әдіспен екі орындық екі предикатты құрылымның семантика және синтаксис қасиеттері анықталған. Әдістеме екі құраманы бір-біріне енгізу жолдарын көрсетеді.*

*In paper author constructed the method of definability of symmetry, irreflexive graphs on the structure with two binary predicates. This method found syntax and semantic conditions of structure with two binary predicates. This method used to solution of problem of realization of structures.*

*Предварительные сведения*

Выбор направления исследования обусловлен тем, что изучение структурных и вычислительных свойств алгебраических объектов посредством понятия определимости является одним из мощных методов в теории вычислимости и теории моделей. Исследование посвящено одному из интенсивно изучаемых областей в этом направлении, а именно проблемам определимости и алгоритмической сложности отношений над алгебраическими структурами.

Всякий раз, как только появляются интересные с точки зрения вычислимости результаты, возникают естественные вопросы о существовании структур с такими же свойствами в известных алгебраических классах, как группы, кольца, решетки, графы и т.п.

S.S.Goncharov, V.S.Narizanov, J.F.Knight и др. [1] показали существование структур с интересными вычислительными и синтаксическими свойствами. Аналог данных результатов доказан для классов алгебраических структур, таких как метаабелевы группы простой экспоненты и без кручения, решеток, колец, коммутативных полугрупп, областей целостности.

**Определение.** Пусть дана структура  $A$  сигнатуры  $\sigma$ , и  $\phi(\bar{x})$  — произвольная формула языка данной сигнатуры;  $\phi(\bar{x})$  — список  $k$  различных переменных. Определим  $k$ -местный предикат  $\phi^A[\bar{x}] \iff \{\bar{a} : \bar{a} \in |A|^k, A \models \phi(\bar{a})\}$ , который назовем  $\Theta$ -формульным на структуре  $A$  или определимым  $\Theta$ -формулой, если  $\phi$  является  $\Theta$ -формулой.

**Определение.** Отношение на  $|A|$  назовем формульно определимым на  $A$ , если оно является формульным предикатом.

Пусть даны структуры  $A$  сигнатуры  $\sigma_0$  и  $B$  сигнатуры  $\sigma_1$ . Не уменьшая общности, положим  $|A| \subseteq |B|$ .

**Определение.** Назовем структуру  $B$   $\sigma_1$ -замыканием структуры  $A$ , а структуру  $A$  —  $\sigma_0$ -ядром (ядром) структуры  $B$ , если выполняются условия:

– (а)  $|A| \subseteq |B|$ ;

– (б)  $|A|$  — инвариантное подмножество в  $B$ , т.е. для любого автоморфизма  $f$  структуры  $B$  имеет место условие  $f(|A|) = |A|$ ;