INTEGRATING MATHEMATICS AND INFORMATICS CONTENT KNOWLEDGE IN SPECIALIZED MATHEMATICS TRAINING

Asen Rahnev* & Kosta Garov **
Plovdiv Unversity „Paisii Hilendarski”
Faculty of mathematics and informatics, bul. Bulgaria 236, Plovdiv 4003, Bulgaria
* assen@uni-plovdiv.bg, ** kosgar@uni-plovdiv.bg

ABSTRACT

The paper considers the integration of mathematics and informatics content knowledge in specialized training courses in mathematics. A sample syllabus is proposed for 11th grade students educated in Comprehensive Secondary Schools. A methodological framework for the unit on Geometry and Programming is elaborated.

I. RELEVANCE OF THE ISSUE

Since ancient times, the mathematical science has been studying the **quantitative characteristics** of objects and phenomena in the real world. It offers adequate **mathematical models** for solutions to problems, related to these quantitative characteristics, as they have been set by society, including, and especially through, well specified procedures – **algorithms**. Through the years, the accumulation of such characteristics (**information**) reached such amounts that their storage in terms of formal representation (**data**) and their use became impossible to be implemented without the assistance of relevant technical means – modern **computers**. Moreover, the procedures for processing these data are so sophisticated and time consuming that their application is feasible only with the assistance of this very same technology driven by related **programs**. At the same time, society is well aware of the fact that making high-level decisions meant to improve its governance is infeasible without **mathematical modeling** of the problems and the use of the whole available information (experience), accumulated through the years. That is why the computer (hardware) and the programming (software) industries are among the fastest growing ones at the moment. It is hard to find a field of human activity where computers and software have not occupied a due place. With the mass production of computers and their interconnection within

local and global area computer networks, information has become accessible worldwide. Modern school graduates **cannot pursue a career**, unless they have mastered a **certain minimum of knowledge and skills** related to **computers and computer programs**.

Taking into account the technological development of society, the Ministry of Education and Science of the Republic of Bulgaria has introduced a new curriculum in the secondary schools. This curriculum features a culture and education field, called "Mathematics, Informatics and Information Technologies". It includes Mathematics, Informatics, and Information Technologies which are studied both as compulsory and elective subjects for general and specialized training. The main goal of the specialized training in Mathematics is **the expansion and deepening** of the students' mathematical knowledge. This work proposes a possibility for integration of knowledge in mathematics and informatics, thus facilitating the achievement of the main goal of specialized training. A syllabus has been designed for specialized training in Mathematics, which includes topics that are not covered in the compulsory training. The study of these topics requires the development of computer programs using a programming language to implement the relevant mathematical methods and formulae. In this way, interdisciplinary connections are established between the two subjects: Mathematics and Informatics. We recommend the use of the programming languages **Pascal** or **C++**, which are studied by Bulgarian students as part of their compulsory training in Informatics. It would be practically reasonable to implement the syllabus proposed within the classes for non-compulsory elective or compulsory elective training in Mathematics in the 11th and 12th classes of the secondary schools.

II. SYLLABUS

We propose the course content in Mathematics and Informatics to be covered within the specialized training classes in Mathematics, c.f. Table 1.

№	**Unit**	**New knowledge - hrs**	**Exercises - hrs**
1.	Programming recursive formulae. Rough calculation of square roots. Calculation of "infinite" sums. Application to the approximate calculation of values of the functions **sin x, cos x**, the number π.	**2**	**4**
2.	Fibonacci numbers. Computing "big" Fibonacci numbers. Developing computer programs handling "big" numbers.	**2**	**4**
3.	Approximate solution of equations. Methods of the chords and tangents. Bisection method. General iteration method. Developing	**3**	**5**

	computer programs for approximate solution of equations using the methods under study.		
4.	Determinants. Calculating determinants of second and third order. Systems of linear equations in two and three variables. Cramer's formulae. Developing computer programs for solving systems of linear equations using Cramer's formulae.	**2**	**4**
5.	Matrices. Operations with matrices. Matrix representation of a system of linear equations in **n** variables. Computer programs for solving systems of linear equations by the methods of Gauss and Gauss-Jordan.	**3**	**6**
6.	Polynomials. Calculating polynomials by Horner's method. Operations with polynomials. Developing computer programs for calculating polynomials and operations with polynomials.	**2**	**4**
7.	Approximation of functions. Interpolation polynomial. Programming of Lagrange's and Newton's interpolation formulae. Inverse interpolation. Extrapolation.	**3**	**5**
8.	Tabulation of functions. Developing computer programs for tabulating different functions.	**1**	**3**
9.	Geometry and programming. Essentials of analytic geometry – coordinates of points in a plane, finding the distance between two points, equation of a straight line, mutual position between a point and a plane, mutual position between two straight lines, convexity of a polygon, calculating the perimeter and the area of a polygon, etc.	**3**	**7**
10.	Calculating the area of a curvilinear trapezium. Quadrature formulae. Programming the formulae for rectangles, trapezia and Simpson's formula.	**3**	**3**
	Total number of hours:	**24**	**44**

Table 1.

The total number of hours envisaged for the implementation of the syllabus is 68, which allows the training to take place within the duration of one school year (36 weeks, 2 hours per week).

III. METHODOLOGY OF TEACHING. SAMPLE FRAMEWORK FOR TEACHING THE UNIT ON "GEOMETRY AND PROGRAMMING"

The proposed syllabus shows that the teaching of some knowledge of the so called "Higher mathematics" is also involved. Certain knowledge in linear algebra, analytic geometry, numerical methods and real analysis is covered. The teaching of this content involves the integration of knowledge in programming, as the developed programs are run on a computer and in this way the students can see the direct implementation of the studied methods and formulae. We suggest that a large part of the mathematical knowledge should be offered to be taken for granted. The mathematical apparatus used to prove many of the facts is comparatively complicated for the students, while the development of computer programs will contribute to the faster acquisition of the studied formulae and methods. Here we propose a sample framework for teaching the unit on "Geometry and Programming".

A. Fundamental geometry facts to teach the unit

1. Coordinates of a point. Distance between two points

Let there be three points in a Cartesian coordinate system: $A_1(x_1,y_1)$, $A_2(x_2,y_2)$ and $C(x_2,y_1)$. The distance between the points A_1 and A_2 is calculated by using the formula:

(1) $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

The DD function in the "Pascal" programming language calculates the length of a segment using the coordinates of its end points:

```
function DD(x1,y1,x2,y2:real): real;
begin
 DD:=sqrt(sqr(x2-x1)+sqr(y2-y1));
end;
```

2. Equation of a line in a plane

We consider line **a**, defined by points $A_1(x_1,y_1)$ and $A_2(x_2,y_2)$. We designate $F(x,y)=(x-x_1)(y_2-y_1)-(y-y_1)(x_2-x_1)$. The equation of the line **a** is $F(x,y)=0$ i.e. $(y_2-y_1)x+(x_2-x_1)y+ y_1(x_2-x_1)- x_1(y_2-y_1)=0$. Let $A=y_2-y_1$, $B=x_1-x_2$, $C=y_1x_2- x_1y_2$, hence the equation of the line A_1A_2 is:

(2) **a**: $Ax+By+C=0$, $A,B,C \in R$.

The point with coordinates $(p,q)\in$**a** $\Leftrightarrow F(p,q)=0$. This reasoning is not valid if the line A_1A_2 is parallel to one of the axes Ox or Oy. When **a**//Oy, the equation of the line **a** is: **a: x=x1**, and when **a**//Oy, then the equation of the line **a** is: **y=y1;**

The **Line** procedure finds the general equation of the straight line **a**.

```
 Procedure Line(x1,y1,x2,y2:integer; Var A,B,C:real);
 begin
 if (x1<>x2) and (y1<>y2) then
              begin A:=y2-y1;B:=x1-x2;C:=x2*y1-x1*y2; end
              else    if (x1=x2) and (y1=y2) then begin
            write('Indefinite straight line');exit;end
            else if x1=x2 then beginA:=1;B:=0;C:=-x1;end
            else if y1=y2 then begin A:=0;
                                     B:=1;
                                     C:=-y1;
                               end;
write ('The equation of the straight line is: ',A,'x+',B,'y+',C,'=0');
end;
```

3. Mutual position of a point and a straight line

The line **a**, specified by the points $A_1(x_1,y_1)$ and $A_2(x_2,y_2)$, divides the plane into two half-planes. The points $P_1(p_1,q_1)$ and $P_2(p_2,q_2)$, not lying on **a,** lie in the same half-plane bounded by the line **a** : Ax+By+C=0, if the expressions $q_1-\frac{-Ap_1-C}{B}$ and $q_2-\frac{-Ap_2-C}{B}$ have the same sign. The points $P_1$and P_2 lie in different half-planes bounded by the line **a,** if these expressions have opposite signs. The **SGNI** function determines the sign of the expression: $q-\frac{-Ap-C}{B}$.

```
Function SGNI(A,B,C,p,q:real):integer;
Var S:real;
begin
  S:=q-(-A*p-C)/B;
  If  S<0 then SGNI:=-1
        else If S=0 then  SGNI:=0
                 else   SGNI:=1;
 end;
```

The **Wz_Pol** procedure determines the mutual position of the straight line A_1A_2 and the two points $P_1(p_1,q_1)$ and $P_2(p_2,q_2)$:

```
Procedure Wz_Pol(A,B,C,p1,q1,p2,q2:real);
 Var f:integer;
```

```
 Begin
  f:=0; p:=p1; q:=q1; f:=f+SGNI(A,B,C,p,q);
        p:=p2;  q:=q2; f:=f+SGNI(A,B,C,p,q);
         if ABS(f)=2 then write('the points lie in the same half-plane bounded by a')
              else write('the points lie in different half-planes bounded by a');   end;
```

4. Polygons. Convexity of geometric figures

In geometry, a **simple polygon** is defined as closed polygonal chain of line segments that do not cross each other[1]. A simple polygon whose interior is a convex set, is called **convex**[2].

Let M be a polygon defined by the coordinates of its vertices M_1, M_2, ...M_n. We designate $M_{n+1}=M_1$. To verify whether M is a convex polygon, the following algorithm can be used:

For any vertex M_i ($1 \le i \le n$), the following operations are carried out:

- the general equation of the line M_iM_{i+1} is elaborated;
- verification is made whether the vertices, other than M_i and M_{i+1}, lie in the same half-plane with respect to the line M_iM_{i+1}. This verification has to be performed for the points from number 1 to i-1 and from i+2 to n.

The IZP function verifies whether a polygon is convex

```
Function IZP(Var X,Y:Masiv; n:integer):boolean;
Type Masiv=array[1..100] of real;
Var s,i,k:integer;
begin IZP:=true; s:=0;
for i:=1 to n do
   begin  Line(X[i],Y[i],X[i+1],Y[i+1],A,B,C);
for k:=1 to n do
         if (k<>i) and (k<>i+1) then s:=s+SGNI(A,B,C,X[k],Y[k]);
         if ABS(S)<>n-2 then begin IZP:=false; exit;
                                        end;
   end;
end;
```

5. Distance from a point to a straight line

Let on a Cartesian plane **Oxy** there be line a: **Ax+By+C=0** and point P(p,q), which does not lie on the line a. The distance from point **P** to the line **a** is determined by the formula:

(3) $$\delta = \frac{|Ap + Bq + C|}{\sqrt{A^2 + B^2}}$$

The **R** function determines the distance from point **P** to the line **a**: Ax+By+C=0

```
Function R(A,B,C,p,q : real) : real;
  begin R:=ABS(A*p+B*q+C)/Sqrt(Sqr(A)+Sqr(B);
end;
```

Using formula (1), one can solve the problem of finding the perimeter and area of a simple polygon defined by the coordinates of its vertices. Let there be the polygon $A=A_1A_2...A_n$, $A_i(x_i,y_i)$, i=1,2,...,n. We assume $x_{n+1} =x_1$,$y_{n+1} =y_1$.The perimeter A is calculated as a sum of the lengths of its sides using the **Length1** function.

```
Function Length1(Var X,Y :Masiv; n: integer):real;
Var i: integer; P1:real;
  begin P1:=0;
    For i:=1 to n do P1:=P1+D(X[i],Y[i],X[i+1],Y[i+1]);
Length1:=P1;
end;
```

The area of the polygon **A** can be determined by adding the areas of the triangles $A_1A_2A_3$, $A_1A_3A_4$,...,$A_1A_{n-1}A_n$. This approach, however, involves a certain loss of precision in cases of big **n**, resulting from the multiple application of the built-in function **Sqrt.** Therefore, we recommend using the formula:

$$\textbf{(4)}\ S_M = \frac{1}{2}\left|\sum_{i=1}^{n}(x_i - x_{i+1})(y_i + y_{i+1})\right| .$$

The **SM** function calculates the area of a simple polygon by the specified vertex coordinates.

```
function SM(Var X,Y:Masiv ; n:integer);real;
  Var I :integer; SM1:real;
   Begin
      SM1:=0.0;
        for i:=1 to n do SM1:=SM1+(x[i]-x[i+1])*(y[i]+y[i+1]);
       SM:=0.5*ABS(SM1);
   end;
```

Using the **SM** function, it can be verified whether the point **P(x,y)** lies on the interior of a simple polygon. To solve this problem, we resort to the following assertion: a point belongs to the interior of a simple polygon, if the polygon area equals the sum of the areas of the triangles obtained from connecting that point with the polygon vertices.

B. Sample solution of a geometry problem using a computer program

Employing the formulae described above and the programming fragments, one can solve even more sophisticated problems. The problem below has been assigned to participants in an informatics competition on a national level.

Problem. Let A and B be convex polygons in a plane with vertices $A_1,A_2,. . .,A_m$ and $B_1,B_2,...,B_n$, respectively, whose coordinates are positive integers (n≥5, m≥5). The arrangement of polygons A and B is such, that vertices $A_2,A_3,...,A_k$ are internal points to polygon B, while vertices $B_2,B_3,...,B_l$ are internal points to polygon A ($1<\kappa<m$, $1<l<n$), as $A_1\equiv B_{l+1}$, $B_1\equiv A_{k+1}$. The vertices of both polygons $A_1,A_2,...,A_m$ and $B_1,B_2,...,B_n$ are arranged counter-clockwise. Develop a program that inputs the vertex coordinates for polygons A and B, produces a visual representation of the polygons on the screen and calculates the areas of polygons A and B, as well as the area of their union, i.e. the polygon with vertices $B_{l+1},B_{l+2},..,A_{k+1},A_{k+2},..,A_m$.

Solution:

1. Values for m and n are entered, as well as coordinates of the vertices of polygons A and B and the values for κ and l ($1<k<m$, $1<l<n$).

2. The areas of the polygons A and B are calculated.

3. The areas of polygons A and B are added up and then the area of the polygon with vertices $B_1B_2...B_lA_1A_2A_3...A_k$ is deducted from that sum.

```
Program Polygon;
 Uses Crt, Graph;
  Type Masiv=array[1..101] of integer;
  Var AX,AY,BX,BY,TX,TY:Masiv;
   GD,GM,i,m,n,l,k :word;
     s:real;
      Procedure Draw(Var X,Y:Masiv; n:integer);
       Var Poly:array[1..101] of PointType;
        i:integer;
          begin
           for I:=1 to n do
                begin Poly[i].x:=X[i]; Poly[i].y:=Y[i];
                 end;
                   Poly[n+1]:=Poly[1];
                    DrawPoly(n+1,Poly);
          end;
  begin { main }
  Clscr;
    repeat write('m=');readln(m);until m>=5;
     repeat write('n=');readln(n);until n>=5;
        repeat write('k='); readln(k); until(k>1)and(k<m);
           repeat write('l='); readln(l);until (l>1)and(l<n);
  writeln('Input coordinates of polygon A:');
    for I:=1 to m do
        begin
          write('x(',i,')='); readln(AX[i]);
          write('y(',i,')='); readln(AY[i]); end;
            AX[m+1]:=AX[1];AY[m+1]:=AY[1];
 writeln('Input coordinates of polygon B:');
```

```
for I:=1 to n do
    begin
        write('x(',i,')='); readln(BX[i]);
        write('y(',i,')='); readln(BY[i]); end;
        BX[n+1]:=BX[1];BY[n+1]:=BY[1];
            GD:=detect;;InitGraph(GD,GM,'');
                Draw(AX,AY,m);
                Draw(BX,BY,n); readln;
        CloseGraph;
writeln('The area of polygon A is=',SM(AX,AY,m):7:2);
 writeln('The area of polygon B is=",SM(BX,BY,n):7:2);
   S:=SM(AX,AY,m)+SM(BX,BY,n);
       for I:=1 to l do
         begin
           TX[i]:=BX[i];
           TY[i]:=BY[i];
         end;
     for i:=l+1 to k+l do
         begin
          TX[i]:=AX[i-l];
          TY[i]:=AY[i-l];
         end;
          TX[k+l+1]:=TX[l]; TY[k+l+1]:=TY[l];
          S:=S-SM(TX,TY,k+l);write('The area of the union of the
two polygons is=',C:7:2);
 end.
```

<u>C. Self-study problems related to the topic</u>

1. Let points $A(a_1,a_2)$, $B(b_1,b_2)$ and $C(c_1,c_2)$ be such that they do not lie on a same line. Develop a program for finding the general equation of the perpendicular bisector of segment AB and the coordinates of the origin O of the circle defined by A, B and C.

2. Let there be n points defined by their coordinates in the plane. Develop a program which determines the width of the narrowest sector containing the points and the origin of a minimum radius circle incorporating all points.

3. Let there be points $A,P_1,P_2,...,P_n$, $n\geq3$, defined by their coordinates in the plane. Develop a program which verifies whether there exists a line that passes through A and divides the plane into two half-planes so that all other points $P_1,P_2,...,P_n$ lie only in one of the half-planes.

4. Let $M_1(x_1,y_1)$, $M_2(x_2,y_2)$, ..., $M_n(x_n,y_n)$ $4\leq n\leq20$ be n points in the plane, whose coordinates $x_1,y_1,x_2y_2,...,x_n,y_n$ are defined as sequential elements of a one-dimension array A. It is known that, given a suitable arrangement,

these points are sequential vertices of a convex polygon. Develop a program which inputs the values for n and the coordinates of the n points, and which:

a) re-arranges the elements on array A of the kind $(x_{i1},y_{i1},x_{i2},y_{i2},...,,x_{in},y_{in})$, so that $M_{i1},M_{,i2},...,M_{in}$ be a convex polygon where x_{ik},y_{ik} are the coordinates of point M_{ik} ;

b) calculates the area of the convex polygon $M_{i1},M_{2},...,M_{inM}$ defined in a).

5. Let there be n positive numbers $\{a_i\}$ i=1,2,...,n. Does there exist a convex n-sided polygon with side lengths $\{a_i\}$, i=1,2,...,n. Is a change in the solution required in case of a simple polygon?

6. Let there be a list of rectangles in the plane, defined by the coordinates of their vertices. Their sides are parallel to the coordinate axes. Develop a program which prints the coordinates of a point which belongs to all rectangles, in case such a point exists.

More problems related to this topic are available in [1] on pages 305-306, in [2] on pages 179-184 and pages 284-285, in [3] on pages 68-78, in [4] on pages 114-122 and in [5] on pages 40-57.

This framework can be applied in a similar fashion to the other topics on the proposed syllabus. For most of the topics, this has already been done in [4] using the BASIC programming language.

NOTES

[1] http://en.wikipedia.org/wiki/Simple_polygon (accessed in March, 2009)

[2] http://en.wikipedia.org/wiki/Convex_and_concave_polygons (accessed in March, 2009)

Acknowledgements: This paper has been partially supported by the Scientific Research Fund at Plovdiv University, Contract IS-M-4/2008.

REFERENCES

1. Azalov, P., F. Zlatarova, Informatics with Pascal, Sofia, Prosveta, 1994 (in Bulgarian)
2. Azalov, P., F. Zlatarova, Informatics with Pascal though examples, quizzes and problems. Sofia, ASIO Publishing House, 1995 (in Bulgarian)
3. Gavrailov, O., K. Garov, Informatics – introductory course for the comprehensive secondary schools, Sofia, ASIO Publishing House, 1994 (in Bulgarian)
4. Garov, K., A. Rahnev, Programming in BASIC - course-book for the elective courses in mathematics in the 9th and 10th grades of the Comprehensive Secondary Schools, Sofia, Ministry of Education, 1986 (in Bulgarian).
5. Rahnev, A., K. Garov, O. Gavrailov, BASIC through problems, Sofia, ASIO Publishing House, 1995 (in Bulgarian)