A program for generating tex source files producing graphic output

Victor N. Fomenko[a], Victor V. Garbaruk[b], Nina V. Popova[c]

[a] LLC "New Technologies and Service" St. Bolshaya Raznochinnaya, house 14, lit. A, room 101, *Saint-Petersburg, 197110, Russia*
[b] *Emperor Alexander I State Transport University, Moscow avenue 9, Saint-Petersburg, 190031, Russia*
[c] *Peter the Great St. Petersburg Polytechnic University, Polytechnicheskaya st., 29,*
St. Petersburg, 195251, Russia

Abstract

In this work a package of applications is described that is aimed to enhance graphic facilities of the Latex typesetting system [1, 2]. The package is built up by the fusion of the Latex formatting and graphic tools, on the one hand, and flexibility and computational aids of FORTRAN 95 [2], on the other. As a result, the program allows to easily create pictures containing many repeating elements or elements varying in a regular manner, automatically hatching a selected domain, inserting formatted objects (a formula, for instance) at any point of the image. The package enables us to create 3D object images as seen along a certain direction. An automatic drawing of their plane sections may be also done.

Keywords [1]
computer graphics, Latex, 3D graphics, Fortran.

1. Introduction

Creating pictures, using WORD also, is not a simple task. LaTeX [1, 2, 3] is a popular TeX typesetting package that makes it easy to type complex documents. The package allows you to automate many tasks of typing books, educational materials, scientific articles. In distance learning LaTeX is actively used to create multivariate assignments in various disciplines [4, 5]. Most documents created by means of LaTeX also contain graphics and illustrations for better understanding of the text. Adding visual elements to text can be done in different ways.

There are many different charting and data visualization systems that can be saved in PDF format. These objects are then included in the text of the document prepared by LaTeX [6].

Another approach is to convert images and graphs from powerful mathematical software packages Matlab [7], Mathematica [8], GeoGebra [9] and many others into "LaTeX-code".

There are a number of packages working well with LaTeX. We list some of these packages, but a detailed discussion of their merits and demerits is beyond the scope of this article. MetaPost [10] is a graphical drawing editor for LaTeX. The package is based on the METAFONT [11] program created by Donald Knuth for the development of vector fonts for TeX. The Asymptote package [12] overcomes some of the MetaPost limitations. The LaTeXDraw package [13] can be used to generate PSTricks code or directly create PDF pictures. PGF / TikZ is a tandem of languages for producing vector graphics from geometric and algebraic descriptions. PGF is a lower-level language, while TikZ is a set of higher-level

Models and Methods for Researching Information Systems in Transport, December 11–12, 2020, Saint Petersburg, Russia
EMAIL: vfomenko1943@gmail.com (Fomenko); vvgarbaruk@mail.ru (Garbaruk); ninavaspo@mail.ru (Popova)
ORCID: 0000-0001-6185-9343 (Fomenko); 0000-0002-9983-4712 (Garbaruk); 0000-0003-3058-7386 (Popova)

macros that use PGF [13-20]. The close relationship between TikZ and PGFPlots allows, firstly, to effectively integrate charts into various drawings made in TikZ, as well as to use the TikZ capabilities when working with charts [20].

In this work software is described which is aimed to make it easier (or, in many situations, even possible at all) to produce complicated images for those who use Latex system. Because of this reason the graphic tool under consideration can prove to be very useful. In this package, the user should write a program in the Fortran 95 language [21]. If the figure is inserted into the text, then the Fortran 95 program ends with subroutine final, and subroutine prep (fn, dims, hrange, vrange, nh, nv, margin) listing the insert parameters begins. Otherwise, the result is obtained in the PDF form.

An important feature of the present way to explore the Latex graphics is performed using facilities of the flexible and well-developed language FORTRAN 95, rather than within the Latex system itself. This language has a library of numerous built-in functions, in particular those which deal with strings, what is important for making files of the tex type. In contrast to this, the graphic applications developed recently are often based on programming within the Latex system which is less capable of implementing algorithms, especially computational ones.

2. Brief description of graphics utilities

We will give a brief description of the utilities (their total number is more than 80) that can be used to create drawings. First of all, the size of the picture is set, then the ranges for the auxiliary coordinates along the horizontal and vertical axes are indicated, the coordinate axes are displayed, they can be provided by numbered divisions and additional marks. Note that the coordinates of the points in the displayed image are set in arbitrary units. The horizontal and vertical scale is determined basing on the size of the picture and the range of coordinates in arbitrary units along the horizontal and vertical lines. In most cases, the lines forming the picture can be solid or dashed, and the dash-size parameters can be set by the user. Line width can also be user-defined.

3. Creation of drawings

In this package, the user has to write a Fortran 95 program. The input file created for the LaTeX system is processed by the LaTeX translator into the PDF file shown in the figures below. We note the following features of the figures constructed using the described software package:

- automatic hatching over a domain with the boundary defined parametrically (fig. 1);
- images of flat pictures with repeating elements, for example, electrical circuits (fig. 2, 3).

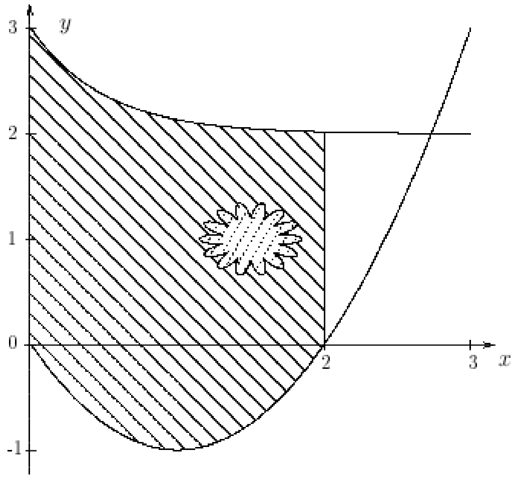

Figure 1: Hatching

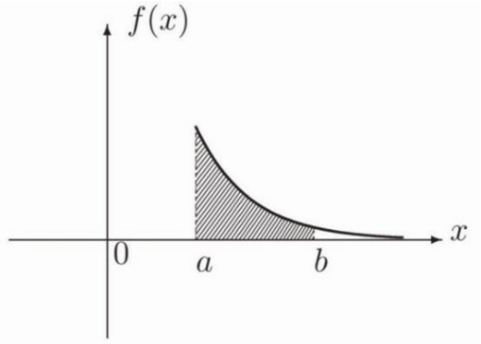

Figure 2: Grafik

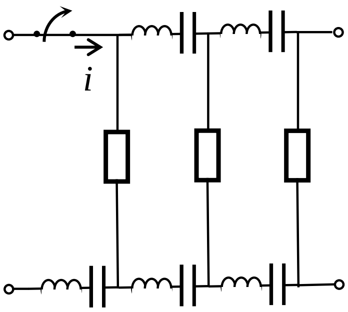

Figure 3: Electrical circuit

4. Imaging of 3D objects

The algorithm implemented in this work corresponds to the photographic process of a removed object if the size of the latter is much less than the distance between it and the camera.

In particular, it is possible to display spatial curves and convex polyhedrons. The direction in which the object is viewed is set by vector (α, β, γ) defining the position of the screen. To obtain, for example, an isometric projection, you may take $\alpha = 1/3$, $\beta = 1/3$, $\gamma = 1/3$.

The position of the camera with respect to its axis is chosen so that vertical elements are depicted vertically as well.

To picture convex polytopes one should indicate the Descartes, cylindrical or spherical coordinates of the vertices. Besides one should specify for each face the vertices which lie in it. The program determines which vertices are connected with edges. Also the edges are found out which are non-visible for the given viewing direction. Such edges are imaged by dash lines. An automatic denotation of vertices is available too. Given the equation of a plane, the program builds the cut of the polytope by this plane.

Note some features of the images produced by described software:

- authenticity, *i.e.*, exact reproduction of geometrical properties up to the size of pixels depicting the image;
- imaging of 3D objects and their plane sections in strict accordance with the rules of orthographic projection.
- image and shading of flat sections of a spatial figure;
- rendering of features after rotation about any spatial axis.

Figure 3 shows a regular 15-faced polyhedron. Below are fragments of the code that provides its 2D image.

1) Specifying the coordinates of vertices of the polytope.

```
n=7
do k=1,n
V(:,k)=(/cos(2*pi/n*k),sin(2*pi/n*k),0./)
V(:,k+n)=(/cos(2*pi/n*k),sin(2*pi/n*k),1./)
enddo
V(:,2*n+l)=(/0.,0.,2.3/)
```

2) Specifying faces by indicating three vertices.

```
do k=1,n
f_index(:,k)=(/k,mod(k,n)+1,k+n/)
enddo
do k=1,n
f_index(:,n+k)=(/2*n+1 ,n+k,n+mod(k,n)+1/)
enddo
f_index(:,2*n+1)=(/1,2,3/)
```

3) Setting the cutting plane.

```
T=(–0.5,1.,0.5)
```

4) Setting a vector that determines the direction of observation.

```
 SCREEN=(/-1.,0.6,-0.8/)
```

5) Image of vertex names.

```
do k=1,n
write(ind,'(i2)')k
if  (k/=3.and.k/=5)  then
uhr=5
else if   (k==3)  then
uhr=2
else
uhr=9
endif
call put(V(:,k),'$A_{'//trim(adjustl(ind))//'}
$',
d=5.,
uhr=uhr)
call
put(V(:,k+n),'$B_{'//trim(adjustl(ind))//'}
$',d=5.,
uhr=uhr)
enddo
```

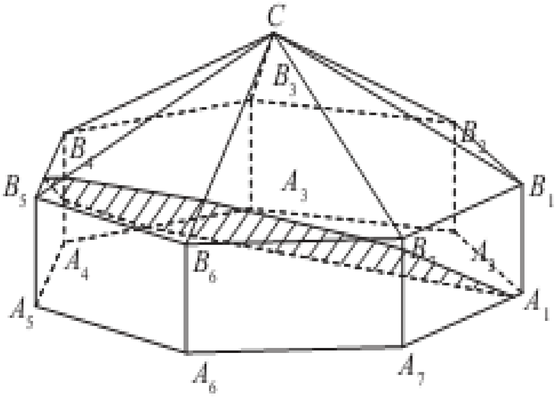

Figure 4: Polytope

The constructed spatial figure can be viewed from various sides by changing the vector that determines the direction of observation. Fig. 5, 6 show top and side views of the previously built Polytope. The parameters of the observation vectors SCREEN = (/0.,0.,-1./) and SCREEN = (/-1.,0.,0./) correspond to these figures.

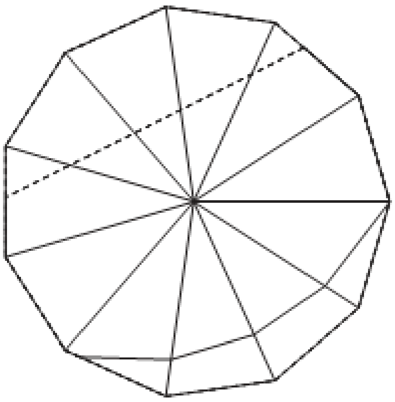

Figure 5: View from above

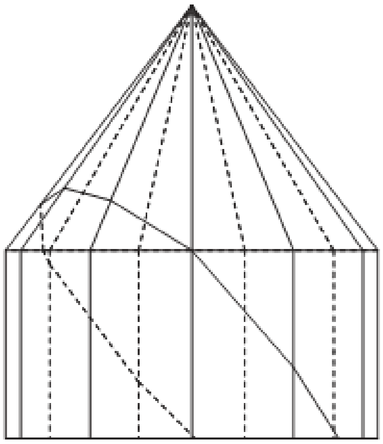

Figure 6: Side view

You can also get an image of a spatial object after rotation around any spatial axis, specified by the direction cosines cx, cy, cz, by an arbitrary angle θ. After the rotation, the point with coordinates x, y, z moves to the point with the coordinates x', y', z' [22],which is calculated by the formula

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} x + \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} y + \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} z,$$

where $\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \lambda_0{}^2 + \lambda_1{}^2 - \lambda_2{}^2 - \lambda_3{}^2 \\ 2(-\lambda_0\lambda_3 + \lambda_1\lambda_2) \\ 2(\lambda_0\lambda_2 + \lambda_1\lambda_3) \end{pmatrix};$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 2(\lambda_0\lambda_3 + \lambda_1\lambda_2) \\ \lambda_0{}^2 - \lambda_1{}^2 + \lambda_2{}^2 - \lambda_3{}^2 \\ 2(-\lambda_0\lambda_1 + \lambda_2\lambda_3) \end{pmatrix};$$

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} = \begin{pmatrix} 2(-\lambda_0\lambda_2 + \lambda_1\lambda_3) \\ 2(\lambda_0\lambda_1 + \lambda_2\lambda_3) \\ \lambda_0{}^2 - \lambda_1{}^2 - \lambda_2{}^2 + \lambda_3{}^2 \end{pmatrix}.$$

Rodrigo-Hamilton parameters $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ are calculated by the formulas:.

$$\lambda_0 = \cos(\theta/2), \;\; \lambda_1 = c_x \sin(\theta/2), \cdot$$

$$\lambda_2 = c_y \sin(\theta/2), \lambda_3 = c_x \sin(\theta/2) \cdot$$

This subroutine is useful for analyzing spatial curves. Figure 7 shows different types of a conical helix defined by the following parametric equations:

$$\begin{cases} x(t) = t \cdot \cos t; \\ y(t) = t \cdot \sin t; \\ z(t) = t. \end{cases}$$

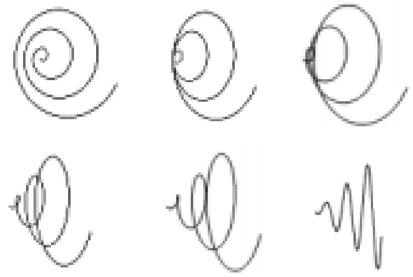

Figure 7: Curves in space

5. Plotting functions

The main disadvantage of the PICTURE graphic package and its EPIC and EEPIC extensions used in the LaTeX system is their orientation towards manual work. For example, they allow you to display a graph, but it is assumed that the coordinates of the points will be entered by the programmer as command arguments. In this package, the user only has to write a Fortran 95 program that defines the curve parametrically. To construct a curve, call the corresponding subroutine of the package, and the names of the procedure defining the curve are included as parameters of this subroutine. It is possible, if necessary, to display a large number of illustrating elements (arrows, lines, etc.).

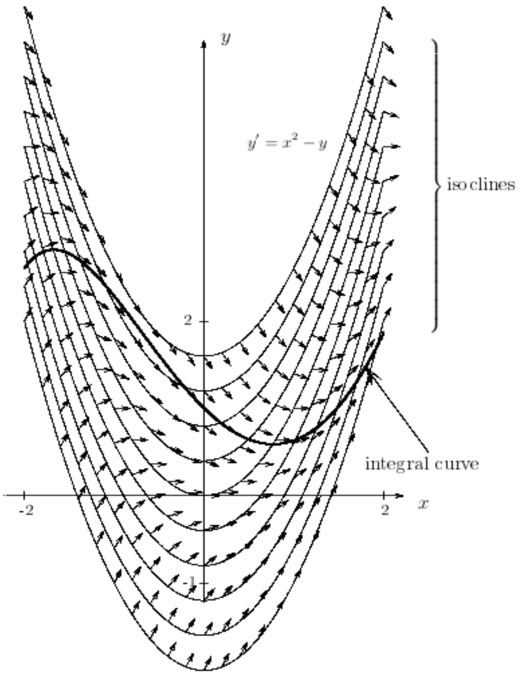

Figure 8: Solution of the equation $y' + y = t$

Figure 8 was created to graphically illustrate the solution of the differential equation $y' + y = x^2$. It shows how the integral curve intersects the isoclines at the direction along the arrows going out from the given isocline. Below are several subroutines that provide the construction of Figure 4.

1) Solution of the differential equation,

```
real function y(t)
y = t**2 – 2*t + 2 + C*exp(–t)
end function y
```

2) Description of isocline (2).

```
real function qy(t)
real t
qy=t**2-C1
end function qy
```

3) display of ten isoclines,

```
N=10
do i=1,N
C1=(Cmax-Cmin)/N*i+Cmin
call curve(-2.,2.,q,qy,1000)
```

4) display of arrows specifying the direction on isoclines,

```
do k=0,kmax
call vector($L1((/q(-2.+4./kmax*k),
qy(-2.+4./kmax*k)/),&atan(C1),4.))
enddo
```

5) Display of a partial solution.

```
C = –1.
call curve(–2.,2.,q,qy,1000,width=5)
```

Figure 9 was created to graphically illustrate the solution of the differential equation $y' = \left(\frac{y}{x}\right)^2$.

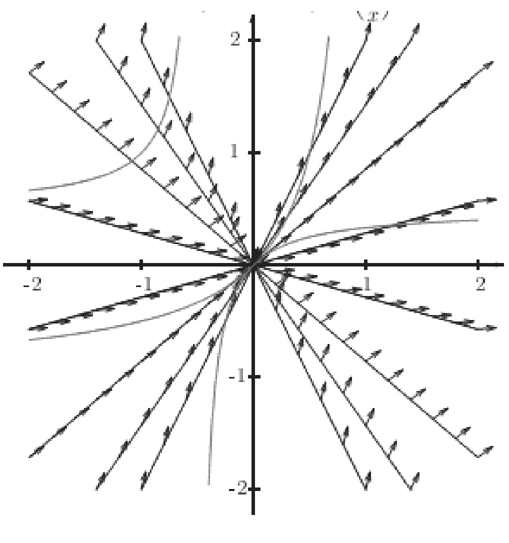

Figure 9: Solution of the equation $y' = \left(\frac{y}{x}\right)^2$

6. Conclusion

An important difference of this approach from using Latex graphics is its hybrid nature. Programming is carried out not within the framework of the Latex system, but by means of the flexible and well-developed language

FORTRAN 95. This language has a rich library of built-in functions, in particular, functions for working with strings, which is especially important for creating tex source files. On the contrary, currently developed graphical applications usually focus on programming in the Latex system itself, which has fewer facilities for implementing algorithms, especially computational ones.

A significant advantage of this approach is its ability to extend itself by creating new procedures based on those already written in Fortran.

References

[1] Goossens, M., Mittelbach, F., Rahtz S., Voss, H. The LaTeX Graphics Companion, Second Edition, New York, Addison-Wesley, 2008.

[2] Spivak M. The Joy of TEX. A gourmet guide to typesetting with the AMS-TEXmacro package. American Mathematical Society, Provi-dence, RI, 1990.

[3] Rozhenko A. I. Art of text composition in LATEX, Printing house of the State University, Novosibirsk, 2005.

[4] V.M. Karnaukhov LATEX application. Generator of test options. Saarbrücken, Germany, Lap lambert Academic Publishing, 2012.

[5] Lubyagina E.N., Yurlova E.S. Preparation of educational assignments in LATEX. (2014) Mathematical bulletin of pedagogical universities and universities of the Volga-Vyatka region. No. 16. 289-295.

[6] Baldin E. M. LaTeX computer printing house, BHV- Petersburg, St.Petersburg, 2012.

[7] Gilat A., Subramaniam V. Numerical methods for engineers and scientists: an introduction with applications using MATLAB. 3rd edition. John Wiley & Sons, Inc. New York, 2014.

[8] Taranchuk V. B. Introduction to graphics of the Mathematica system, BSU, Minsk: 2017.

[9] Kovács, Z., Parisse B. Giac and GeoGebra: improved Grobner basis computations, RICAM Institute, Linz, 2013.

[10] Hobby J.D. A user’s manual for METAPOST. Technical Report 162, AT&T Bell Laboratories, Murray Hill, New Jersey, 1992.

[11] Knuth D. The METAFONTbook. Addison Wesley, Boston, 1986.

[12] Asymptote: The Vector Graphics Language.
URL: http://asymptote.sourceforge.net.

[13] http://latexdraw.sourceforge.net

[14] Kuretenko U. A. TikZ & PGF. Grafiics of Document LaTeX. Rostov-Don, 2014. URL:http://elib.ict.nsc.ru/jspui/handle/ICT/1488.

[15] Tantau T. TikZ and pgf manual. Version 1.01. URL:
http://sourceforge.net/projects/pgf/.

[16] Mertz A., Slough W. Graphics with PGF and TikZ. TUGboat, Volume 28, No. 1 (2007) 91-99, Proceedings of the Practical TEX 2006 Conference. URL: http://works.bepress.com/william_slough/5/.

[17] Feuersanger C. Manual for Package pgfplots - 2D/3D Plots in LATEX. Vers. 1. URL:
http://sourceforge.net/projects/pgfplotsDr.

[18] Esajan A. R., Chubarikov V. N., Djbrovolski N. M., Jakushin A. V. Plotting using LaTeX-package pgfplots, izd-vo TGPU, Tula, 2015.

[19] Plotting with TikZ and LaTeX.
URL: http://pgfplots.net.

[20] Getreuer P. TikZ for High-Quality LaTeX Pictures. URL:
http://www.sciweavers.org/tutorials/tikz-high-quality-latex-pictures.

[21] Chapman, S.J. Fortran 90-95 for Scientists and Engineers, WCB/McGraw-Hill Higher Education, New York, 1998.

[22] Lurie A.I. Analytical Mechanics, Springer, Berlin, 2002.