

# A Perl port of the mathsPIC graphics package

*Dedicated to the fond memory of Mikhail Syropoulos, my beloved brother.*  
— Apostolos Syropoulos

Apostolos Syropoulos  
Department of Civil Engineering  
Democritus University of Thrace  
Xanthi, Greece  
apostolo@obelix.ee.duth.gr

Richard W. D. Nickalls  
Department of Anaesthesia  
City Hospital, Nottingham, UK  
dicknickalls@compuserve.com

## Abstract

This article describes the authors' experience of porting the mathsPIC graphics package to Perl. The motivation for using Perl is described, as well as the reasons for developing it using the Noweb literate programming system. Finally, a simple example is presented.

## Introduction

This article is a short *work in progress* report of our porting of the graphics package mathsPIC (CTAN/graphics/pictex/mathspic/) to standard Perl; a project designed to make it available for a wide range of platforms.

MathsPIC (Nickalls, 1999) is a filter program for use with the excellent P<sub>T</sub>CT<sub>E</sub>X drawing engine<sup>1</sup>. MathsPIC differs from other graphics packages in that it provides an environment for manipulating named points, which greatly facilitates the drawing of geometrical figures. It also accommodates relative addressing, scalar variables, and file input for data points and other commands. MathsPIC was originally written in PowerBASIC 3.5, a commercial version of BASIC available only for MS-DOS systems.

However the original MS-DOS version of mathsPIC does have certain limitations; for example, it is unable to parse mathematical functions. Furthermore, maths libraries for PowerBASIC are all commercial. Consequently, in order to significantly extend mathsPIC, the authors felt it was necessary to reimplement it in a systems programming language consistent with the philosophy 'write once, run everywhere', namely Perl.

<sup>1</sup> The original P<sub>T</sub>CT<sub>E</sub>X files have been significantly improved and made memory-efficient by Andreas Schrell—see `pictexwd.sty` (CTAN/graphics/pictex/addon/)

This reimplementing of mathsPIC started in January 2000, and has been conducted as a collaborative project over the internet, by the authors. The authors used different platforms during the development (Perl 5.6 on a Solaris x86 machine, and djgpp Perl 5.005 on an MS-DOS machine).

Probably the most difficult aspects were maintaining a consistent syntax, and downwards compatibility. We decided to maintain case-insensitivity for command names, as this was found to be particularly useful in promoting readability. The log file (`.mlg` file) was structured to mirror the usual T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X log files in order to allow tools which process these files to work similarly with mathsPIC. This was also an appropriate time to revise and improve the syntax of the language. So, for example, the original `variable` command format, e.g., `variable(x){b,advance(4)}`, has been improved to allow an algebraic syntax, e.g., `var x=b+4`, as well as allowing several variables to be defined using one command, e.g., `var y=3*(j-2), j27=r/3, p4=AB`.

## Why Perl?

Perl is a high-level scripting programming language with an eclectic heritage designed by Larry Wall. It is an interpreted language which has been ported to most operating systems, and so is a particularly good choice for the implementation of programs

In this section we define a few global variables. More specifically: variable `$version_number` contains the current version number of the program, variable `$commandLineArgs` contains the command line arguments. These two variables are used in the `print_headers` subroutine. Variable `$command` will contain the whole current input line. Hash `%PointTable` is used to store point names and related information. Hash `%VarTable` is used to store mathsPIC variable names and related information. Variable `$no_errors` is incremented whenever the program encounters an error in the input file. Variables `$xunits`, `$yunits` and `$units` are related to the `paper` command. In particular, variable `$units` is used to parse the unit part of the unit part of the `paper` command. Variable `$defaultsymbol` is used to set the point shape. Variable `$PI` holds the value of the pi constant.

```
<Define global variables>=
$version_number = "0.0 September 1, 2000";
$commandLineArgs = join(" ",@ARGV);
$command = "";
$curr_in_file = "";
keys(%PointTable) = 0;
keys(%VarTable) = 0;
$no_errors = 0;
$xunits = "1pt";
$yunits = "1pt";
$units = "pt|pc|in|bp|cm|mm|dd|cc|sp";
$defaultsymbol = '$\bullet$';
$PI = atan2(1,1)*4;
```

Used above.

**Figure 1:** Literate (woven) extract from `mathsPIC` source

likely to be used on many different operating systems. Its design has been largely influenced by the C programming language, but also by other tools and languages, for example, SED, AWK, and the Unix shell. Perl's process, file, and text manipulation facilities make it particularly well-suited for tasks involving quick prototyping, system utilities, software tools, system management tasks, database access, graphical programming, networking, and world wide web programming. Finally, and importantly, Perl is in the public domain, and so there are no commercial restrictions.

### Literate programming

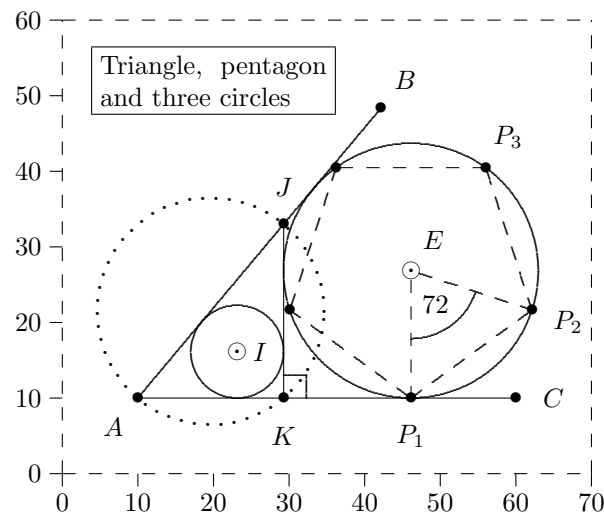
The Perl `mathsPIC` program was developed throughout using so-called 'literate program' methodology (Knuth, 1992; Syropoulos, 1999; Hatzigeorgiu and Syropoulos, 1998), in conjunction with Norman

Ramsey's freely available Noweb literate programming tool (Ramsey, 1994).

Noweb was chosen partly because it is language-independent, but also because it can generate the weaved file in variety of formats, e.g., plain `TEX`, `LATEX`, HTML and `nroff`. Noweb allows programs to be built up of named chunks in any order with documentation interleaved, and has powerful indexing and cross-referencing facilities. Furthermore, Noweb's pipeline makes it easy to extend, and different stages of the pipeline can be implemented in different programming languages (Ramsey, 1994). Noweb uses its `notangle` and `noweave` tools to extract code and documentation as required. Figure 1 shows a formatted code chunk of the resulting literate program.

**Example**

By way of example we show the `mathsPIC` script file (`fig2.m`) which produced Figure 2, and also the associated output `LATEX` file (`fig2.mt`) generated by `mathsPIC`. Note the use of `pictexwd.sty` (CTAN/graphics/pictex/addon/).



**Figure 2:** Example diagram (`mathsPIC` source in text)

```

% mathsPIC script file (fig2.m)
% Figure 2
\documentclass[a4paper]{article}
\usepackage{pictexwd}
\begin{document}
\beginpicture
\setdashes
paper{units(mm),xrange(0,70),yrange(0,60),
      axes(LBT*R*),ticks(10,10)}

\setsolid
point(A){10,10}    % anchor point
point(B){A, polar(50,50 deg)}
point(C){A, polar(50,0 deg)}
point(J){pointonline(AB,30)}
point(K){perpendicular(J,AC)}
drawRightangle(JKC,3)
drawLines(AB,AC,JK)
drawIncircle(AJK)
drawExcircle(AJK,JK)
\setplotsymbol({\large .})
\setdots
drawCircumcircle(AJK)
point(I){IncircleCenter(AJK)}[$\odot$]
point(E){ExcircleCenter(AJK,JK)}[$\odot$,1.2]
point(P1){perpendicular(E,AC)}
var r = EP1    % radius of excircle
var d = 360/5  % angle for pentagon

```

```

var a1=-90,a2=a1+d,a3=a2+d,a4=a3+d,a5=a4+d
point(P2){E, polar(r,a2)}
point(P3){E, polar(r,a3)}
point(P4){E, polar(r,a4)}
point(P5){E, polar(r,a5)}
drawPoints(ABCJKIEP1P2P3P4P5)
\setplotsymbol({\tiny .})
\setdashes
drawLine(P1P2P3P4P5P1,EP1,EP2)
\setsolid
drawAnglearc{angle(P2EP1),radius(9),
              internal,clockwise}
\newcommand{\figtitle}{%
\ \begin{minipage}{30mm}%
\ Triangle, pentagon and three circles%
\ \end{minipage}%
\ }%
text(\fbox{\figtitle}){20,52}
variable(s){5}
text($A$){A, polar(s,230 deg)}
text($B$){B, polar(s,50 deg)}
text($C$){C, polar(s,0 deg)}
text($J$){J, polar(s,90 deg)}
text($K$){K, polar(s,270 deg)}
text($E$){E, polar(s,0 deg)}
text($72$){E, polar(5.5,-54 deg)}
text($I$){I, shift(3, 0)}
text($P_1$){P1, polar(s,a1)}
text($P_2$){P2, polar(s,a2)}
text($P_3$){P3, polar(s,a3)}
\endpicture
\end{document}

```

The axes and bounding box used while constructing the figure are shown here in order to make it easier to understand the `mathsPIC` script file and the output `LATEX` file (`fig2.mt`). They are easily removed simply by commenting out the `axes` and `ticks` options from the `mathsPIC paper` command.

Note that in addition to `mathsPIC` commands (*not* prefixed with a backslash), the script file can also contain `PICTEX`, `TEX` and `LATEX` commands, since `mathsPIC` processes the input file according to the following rules.

- `mathsPIC` commands are converted into their equivalent `PICTEX` commands. The `mathsPIC` commands are also copied verbatim but commented out—this makes the output file easier to understand.
- Lines having a leading backslash followed by *one or more spaces* (e.g., `\_`) are copied verbatim except for the leading backslash.

- Lines having a leading backslash followed by a non-space character (e.g., `\setdashes`) are copied verbatim.

Note that the output L<sup>A</sup>T<sub>E</sub>X file (`fig2.mt`) also contains some additional information (e.g., coordinates of derived points, radius of circles etc.) which is usually included at the end of the original `mathsPIC` command, but sometimes as a separate line. Once the figure is finished, the output file can be generated *without* any comment lines simply by using the `-c` command-line switch during the final `mathsPIC` run.

## References

Hatzigeorgiu, N and A. Syropoulos. “Literate Programming and the ‘Spaniel’ Method”. *SIGPLAN Notices* **33**(12), 52–56, 1998.

Knuth, D E. *Literate programming*. Number 27 in CSLI Lecture Notes. Center for the Study of Language and Information, 1992. Co-published by Cambridge University Press.

Nickalls, R W D. “MathsPIC: a filter program for use with P<sub>I</sub>C<sub>T</sub>E<sub>X</sub>”. In *Proceedings of EuroT<sub>E</sub>X’99*, pages 192–210, Heidelberg, Germany. 1999. A slightly updated version of this paper was published (in English) in the Greek T<sub>E</sub>X Friends’ journal *Eutopon*, No. 3 (October 1999), pp. 33–49.

Ramsey, N. “Literate programming simplified”. *IEEE Software* **11**(5), 97–105, 1994.

Syropoulos, A. “Literate programming: the other side of the coin”. *RAM Magazine* (129), 248–253, 1999. In Greek.

```
%% mathsPIC output file (fig2.mt)
%% Figure 2
\documentclass[a4paper]{article}
\usepackage{pictexwd}
\begin{document}
\beginpicture
\setdashes
%% paper{units(mm),xrange(0,70),yrange(0,60),axes(LBT*R*),ticks(10,10)}
\setcoordinatesystem units < 1mm, 1mm>
\setplotarea x from 0 to 70, y from 0 to 60
\axis left ticks numbered from 0 to 60 by 10 /
\axis right /
\axis top /
\axis bottom ticks numbered from 0 to 70 by 10 /
\setsolid
%% point(A){10,10} ( 10 , 10 ) %% anchor point
%% point(B){A,polar(50,50deg)} ( 42.13938 , 48.30222 )
%% point(C){A,polar(50,0deg)} ( 60 , 10 )
%% point(J){pointonline(AB,30)} ( 29.28363 , 32.98133 )
%% point(K){perpendicular(J,AC)} ( 29.28363 , 10 )
%% drawRightangle(JKC,3)
\plot 32.28363 10 32.28363 13 /
\plot 29.28363 13 32.28363 13 /
%% drawLines(AB,AC,JK)
\plot 10 10 42.13938 48.30222 / %% AB
\putrule from 10 10 to 60 10 %% AC
\putrule from 29.28363 32.98133 to 29.28363 10 %% JK
%% drawIncircle(AJK)
%% Incircle centre = 23.15115 , 16.13248 ; Radius = 6.132483
\circulararc 360 degrees from 29.28363 16.13248 center at 23.15115 16.13248
%% drawExcrcircle(AJK,JK)
%% Excrcircle centre = 46.13249 , 26.84886 ; Radius = 16.84886
\circulararc 360 degrees from 62.98135 26.84886 center at 46.13249 26.84886
\setplotsymbol({\large .})
\setdots
%% drawCircumcircle(AJK)
```

```

%% circumcircle centre = 19.64182 , 21.49067 ; Radius = 15
\circulararc 360 degrees from 34.64181 21.49067 center at 19.64182 21.49067
%% point(I){IncircleCenter(AJK)}{[$\odot$]} ( 23.15115 , 16.13248 )
%% point(E){ExcircleCenter(AJK,JK)}{[$\odot$,1.2]} ( 46.13249 , 26.84886 )
%% point(P1){perpendicular(E,AC)} ( 46.13249 , 10 )
%% var r = EP1 ( 16.84886 ) %% radius of excircle
%% var d = 360/5 ( 72 ) %% angle of pentagon (deg)
%% var a1=-90, a2=a1+d, a3=a2+d, a4=a3+d, a5=a4+d
%% a1 = -90
%% a2 = -18
%% a3 = 54
%% a4 = 126
%% a5 = 198
%% point(P2){E,polar(r,a2)} ( 62.1567 , 21.64227 )
%% point(P3){E,polar(r,a3)} ( 56.036 , 40.47987 )
%% point(P4){E,polar(r,a4)} ( 36.22898 , 40.47987 )
%% point(P5){E,polar(r,a5)} ( 30.10827 , 21.64227 )
%% drawPoints(ABCJKIEP1P2P3P4P5)
\put {$\bullet$} at 10 10 %% A
\put {$\bullet$} at 42.13938 48.30222 %% B
\put {$\bullet$} at 60 10 %% C
\put {$\bullet$} at 29.28363 32.98133 %% J
\put {$\bullet$} at 29.28363 10 %% K
\put {$\odot$} at 23.15115 16.13248 %% I
\put {$\odot$} at 46.13249 26.84886 %% E
\put {$\bullet$} at 46.13249 10 %% P1
\put {$\bullet$} at 62.1567 21.64227 %% P2
\put {$\bullet$} at 56.036 40.47987 %% P3
\put {$\bullet$} at 36.22898 40.47987 %% P4
\put {$\bullet$} at 30.10827 21.64227 %% P5
\setplotsymbol({\tiny .})
\setdashes
%% drawline(P1P2P3P4P5P1,EP1,EP2)
\plot 46.13249 10 62.1567 21.64227 / %% P1P2
\plot 62.1567 21.64227 56.036 40.47987 / %% P2P3
\putrule from 56.036 40.47987 to 36.22898 40.47987 %% P3P4
\plot 36.22898 40.47987 30.10827 21.64227 / %% P4P5
\plot 30.10827 21.64227 46.13249 10 / %% P5P1
\putrule from 46.13249 25.64886 to 46.13249 10 %% EP1
\plot 47.27376 26.47803 62.1567 21.64227 / %% EP2
\setsolid
%% drawAnglearc{angle(P2EP1),radius(9),internal,clockwise}
\circulararc -72 degrees from 54.692 24.0677 center at 46.13249 26.84886
\newcommand{\figtitle}{%
  \begin{minipage}{30mm}%
    Triangle, pentagon and three circles%
  \end{minipage}%
}%
%% text(\fbox{\figtitle}){20,52}
\put {\fbox{\figtitle}} at 20 52
%% variable(s){5} ( 5 )
%% text($A$){A,polar(s,230deg)}
\put {$A$} at 6.786062 6.169777
%% text($B$){B,polar(s,50deg)}

```

```
\put {$B$} at 45.35332 52.13245
%% text({C$}){C,polar(s,0deg)}
\put {$C$} at 65 10
%% text({J$}){J,polar(s,90deg)}
\put {$J$} at 29.28363 37.98133
%% text({K$}){K,polar(s,270deg)}
\put {$K$} at 29.28363 5
%% text({E$}){E,polar(s,0deg)}
\put {$E$} at 51.13249 26.84886
%% text({72$}){E,polar(5.5,-54deg)}
\put {$72$} at 49.36531 22.39926
%% text({I$}){I,shift(3,0)}
\put {$I$} at 26.15115 16.13248
%% text({P_1$}){P1,polar(s,a1)}
\put {$P_1$} at 46.13249 5
%% text({P_2$}){P2,polar(s,a2)}
\put {$P_2$} at 66.91198 20.09719
%% text({P_3$}){P3,polar(s,a3)}
\put {$P_3$} at 58.97492 44.52495
\endpicture
\end{document}
```



Richard W.D. Nickalls and Apostolos Syropoulos