

# The ‘pst-poly’ package

## A PSTricks package for polygons

Denis GIROU\*

Version 1.5

February 19, 2001

Documentation revised February 19, 2001

### Abstract

This package allow to draw easily various kinds of regular or non regular polygons, using the unique macro `\PstPolygon`, with various customization parameters.

It is also a good example of the great power and flexibility of PSTricks, as in fact it is a very short program (it body is only 100 lines long) but nevertheless really powerful.

And last, it is also a good pedagogical example of how to design and program *high level graphic objects* above PSTricks own ones.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>2</b>
2.1	Parameters . . . . .	2
2.2	Pre-defined polygons . . . . .	5
2.3	Non regular polygons . . . . .	6
2.4	Nodes (vertices) . . . . .	6

## 1 Introduction

‘pst-poly’ offer a unique macro (plus some aliases to define some often used polygons) with few parameters to interact on it. But we can also use all the relevant PSTricks parameters to change the size, the characteristics of lines, to add filling, etc.

The polygons are always drawn counter clockwise.

The syntax is simply: `\PstPolygon*[optional_parameters]`

As for PSTricks closed objects, the `*` version use a solid style to fill the polygon, use the line color for fill color and set the linewidth to 0.

---

\*CNRS/IDRIS — Centre National de la Recherche Scientifique / Institut du Développement et des Ressources en Informatique Scientifique — Orsay — France — <Denis.Girou@idris.fr>.

A preliminary version, limited to *regular* polygons, was shown at TUG'95 conference, St Petersburg, Florida, as part of my talk *Building high level objects in PSTricks* [2].

*Note:* You can take advantage to look also at the polygon feature of the  $\text{\texttt{Xy-pic}}$  package [5], [3].

## 2 Usage

### 2.1 Parameters

There are **eight** specific parameters defined to change the way the polygons are defined:

**PstPicture** (boolean) : to define or not a **pspicture** environment for the polygon.

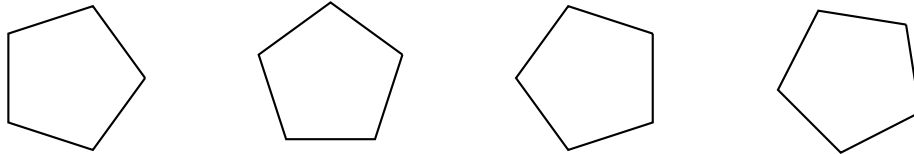
We have to define this parameter to *false* if we want to mix the polygon with other PSTricks objects — see examples later (*Default: true* — which is not the case for basic PSTricks objects).

**PolyRotation** (real) : rotation angle applied to the polygon (*Default: 0* — no rotation).

```

1 \PstPolygon\hfill
2 \PstPolygon[PolyRotation=18] \hfill
3 \PstPolygon[PolyRotation=36] \hfill
4 \PstPolygon[PolyRotation=45]

```

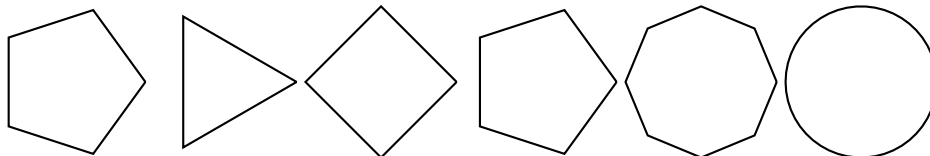


**PolyNbSides** (integer) : number of sides of the polygon (*Default: 5*).

```

1 \PstPolygon
2 \PstPolygon[PolyNbSides=3]
3 \PstPolygon[PolyNbSides=4]
4 \PstPolygon[PolyNbSides=5]
5 \PstPolygon[PolyNbSides=8]
6 \PstPolygon[PolyNbSides=50]

```

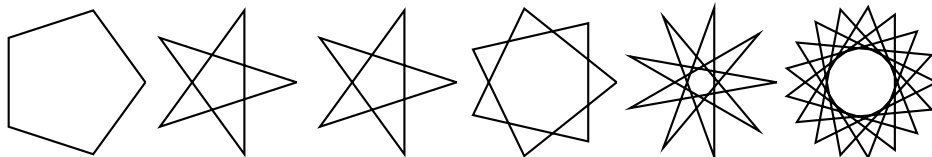


PolyOffset (integer) : number of nodes to bypass to obtain each time the next one  
(Default: 1 — no node bypassed).

```

1 \PstPolygon
2 \PstPolygon[PolyOffset=2]
3 \PstPolygon[PolyOffset=3]
4 \PstPolygon[PolyNbSides=7,PolyOffset=2]
5 \PstPolygon[PolyNbSides=9,PolyOffset=4]
6 \PstPolygon[PolyNbSides=17,PolyOffset=6]

```

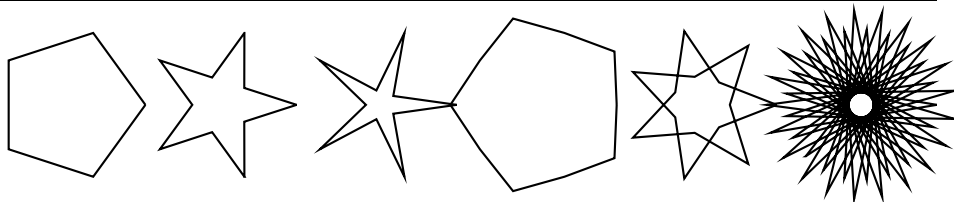


PolyIntermediatePoint (real) : position of the intermediate point used to join each time the next node (Default: empty — not used).

```

1 \PstPolygon
2 \PstPolygon[PolyIntermediatePoint=0.38]
3 \PstPolygon[PolyIntermediatePoint=0.2]
4 \PstPolygon[PolyIntermediatePoint=1.2]
5 \PstPolygon[PolyNbSides=7,PolyOffset=2,
6   PolyIntermediatePoint=0.38]
7 \PstPolygon[PolyNbSides=21,PolyOffset=2,
8   PolyIntermediatePoint=-1.25]

```

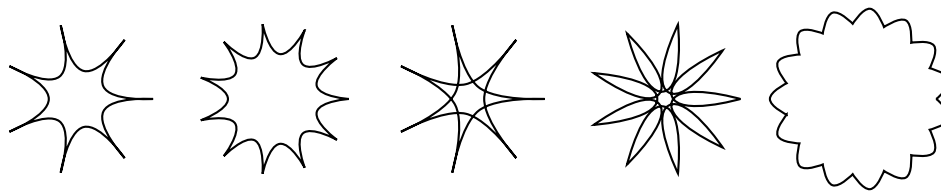


PolyCurves (boolean) : boolean value to choose between straight line and curve to join each time the next node (Default: false — straight lines).

```

1 \psset{PolyCurves=true}
2 \PstPolygon[PolyNbSides=7,PolyIntermediatePoint=0.38]\hfill
3 \PstPolygon[PolyNbSides=11,PolyIntermediatePoint=0.6]\hfill
4 \PstPolygon[PolyNbSides=7,PolyIntermediatePoint=0.2,
5   PolyOffset=2]\hfill
6 \PstPolygon[PolyNbSides=9,PolyIntermediatePoint=0.1]\hfill
7 \PstPolygon[PolyNbSides=15,PolyIntermediatePoint=1.2]

```

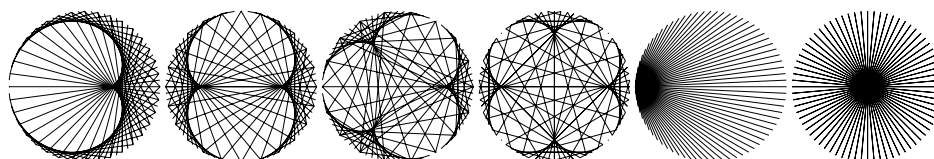


**PolyEpicycloid** (boolean) : boolean value to choose between polygon and epicycloid (*Default: false* — polygon).

```

1 \psset{linewidth=0.001,PolyNbSides=72,PolyEpicycloid=true}
2 % Epicycloid of factor 1 is cardioid and of factor 2 nephroid
3 \multido{\i=2+1}{4}{\PstPolygon[PolyOffset=\i]\hfill}
4 \PstPolygon[PolyOffset=72]\hfill % Epicycloid of factor 71
5 \PstPolygon[PolyOffset=73]      % Epicycloid of factor 72

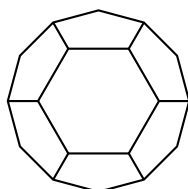
```



**PolyName** (string) : name of the polygon, useful to have different names for the nodes of different polygons (*Default: empty* — no name).

The center of the polygon has name **PolyName0** and the nodes (vertices) have names **PolyName1** to **PolyNameN**.

With this parameter, we can connect as we want nodes of different polygons:

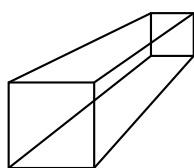


```

1 \psset{PstPicture=false}
2 \begin{pspicture}(-1,-1)(1,1)
3   \PstPolygon[unit=0.8,PolyName=A,PolyNbSides=6]
4   \PstPolygon[unit=1.2,PolyName=B,PolyNbSides=12]
5 \end{pspicture}
6 \multido{\i=1+2}{6}{%
7   \ncline{A}{the\multidocount}{B\i}}

```

It is also a way (limited in fact...) to define three dimensional objects in perspective (this was used first in [5]):



```

1 \psset{unit=0.8}
2 \begin{pspicture}(3,2.5)
3   % \PstSquare is described later
4   \rput[1b](0,0){\PstSquare[PolyName=A]}
5   \rput[1b](2.5,2){\PstSquare[unit=0.5,
6                       PolyName=B]}
7   \multido{\i=1+1}{4}{\ncline{A\i}{B\i}}
8 \end{pspicture}

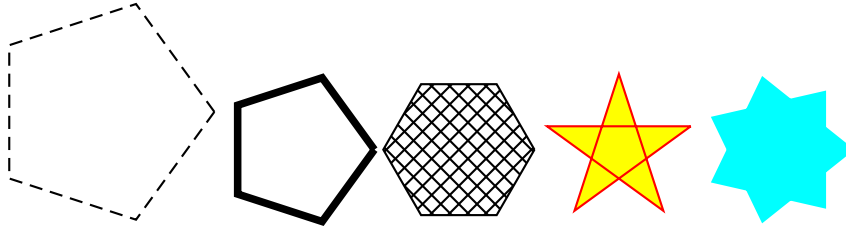
```

Of course, we can mix specific ‘pst-poly’ parameters with relevant PSTricks ones and combine it with other generic macros (for repetitions, projection in the 3d space, etc.)

```

1 \PstPolygon[unit=1.5,linestyle=dashed]
2 \PstPolygon[linewidth=1mm]
3 \PstPolygon[fillstyle=crosshatch,PolyNbSides=6]
4 \PstPolygon[linecolor=red,fillstyle=solid,fillcolor=yellow,
5   PolyRotation=13,PolyOffset=2]
6 \PstPolygon*[linecolor=cyan,PolyNbSides=7,PolyOffset=2]

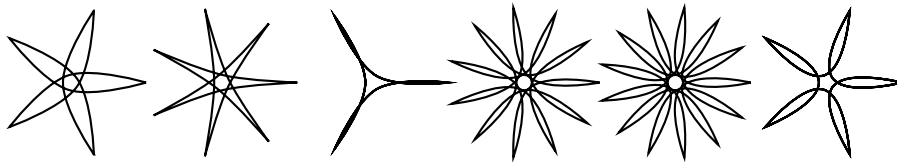
```



```

1 \multido{\i=5+2}{6}{%
2   \PstPolygon[PolyNbSides=\i,PolyCurves=true,
3     PolyIntermediatePoint=0.1,PolyOffset=3] }

```



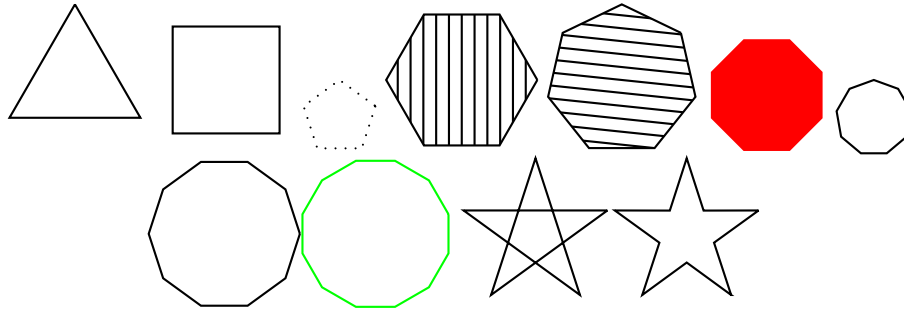
## 2.2 Pre-defined polygons

Some often used polygons and other related geometric objects are pre-defined, for immediate usage :

```

1 \PstTriangle
2 \PstSquare
3 \PstPentagon[unit=0.5,linestyle=dotted]
4 \PstHexagon[fillstyle=hlines,hatchangle=90]
5 \PstHeptagon[fillstyle=vlines]
6 \PstOctagon*[unit=0.8,linecolor=red]
7 \PstNonagon[unit=0.5]
8 \PstDecagon
9 \PstDodecagon[linecolor=green]
10 \PstStarFiveLines
11 \PstStarFive

```



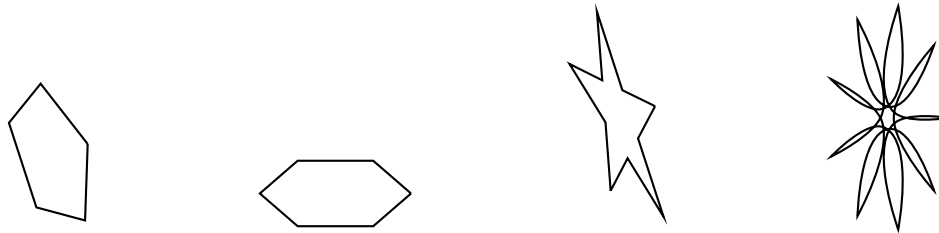
## 2.3 Non regular polygons

Until now, we have described only the so-called *regular* polygons, which are from far the most useful ones (all of them have equal edges and angles). Nevertheless, it is not so difficult to extend these polygon to *non regular* ones, using a different value for horizontal and vertical units (nevertheless, the code is more tricky, as we must do all the trigonometry explicitly...)

```

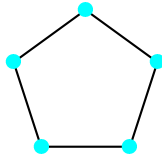
1 \PstPentagon[xunit=0.5]\hfill
2 \PstHexagon[yunit=0.5]\hfill
3 \PstStarFive[xunit=0.5,yunit=1.5]\hfill
4 \PstPolygon[xunit=0.8,yunit=1.5,PolyNbSides=9,PolyOffset=2,
5             PolyIntermediatePoint=0.1,PolyCurves=true]

```



## 2.4 Nodes (vertices)

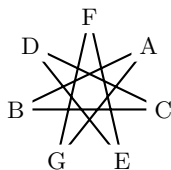
And another powerful possibility is to define a command `\PstPolygonNode` which will be executed at each node (*Default: empty* — nothing executed). The counter name for nodes is `\INode`, starting from 0. The `\multidocount` counter, from the `\multido` command, start itself from 1.



```

1 \providecommand{\PstPolygonNode}{%
2   \psdots[dotsize=0.2,linecolor=cyan](1;\INode)}
3 \PstPentagon

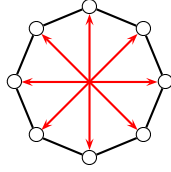
```



```

1 \newcounter{Letter}
2 \providecommand{\PstPolygonNode}{%
3   \setcounter{Letter}{\the\multidocount}%
4   \rput*{*0}(1;\INode){\small\Alph{Letter}}}
5 \PstHeptagon[PolyOffset=3]

```

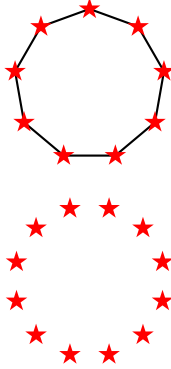


```

1 \providecommand{\PstPolygonNode}{%
2   \psdots[dotstyle=o,dotsize=0.2](1;\INode)
3   \psline[linecolor=red]{->}(0.9;\INode)}
4 \PstPolygon[PolyNbSides=8]

```

It is also a way to nest polygons:



```

1 \newbox{\Star}
2 \savebox{\Star}{%
3   \PstStarFive*[unit=0.15,linecolor=red]}
4 \providecommand{\PstPolygonNode}{%
5   \rput{*0}(1;\INode){\usebox{\Star}}}
6 \shortstack{\PstNonagon\ \ [5mm]
7             \PstDodecagon[linestyle=none]}

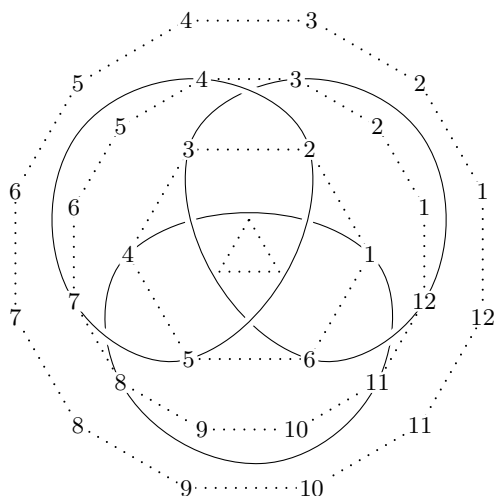
```

Here we redraw one of the polygon examples built with the `Xy-pic` package in [3, example 5-5-34, page 203]:

```

1 \psset{unit=2,PstPicture=false}
2 \begin{pspicture}(-1.6,-1.6)(1.6,1.6)
3   % Just to name the nodes, to be able to join some of them.
4   % by solid curves. We can't draw the polygons now,
5   % as the node numbers must erase the solid curves...
6   \PstHexagon[unit=0.8,linestyle=none,PolyName=H]
7   \PstDodecagon[unit=1.2,linestyle=none,PolyName=D]
8   {\SpecialCoor
9     \psset{linewidth=0.4pt,border=2pt,nodesep=0.45}
10    \psccurve(H1)(H4)(D8)([angle=-30]D9)(D11)
11    \psccurve(D3)(H3)(H6)(D12)([angle=90]D1)
12    \psccurve(D4)(H2)(H5)(D7)([angle=90]D6)}
13 \psset{linestyle=dotted,framesep=1pt}
14 \PstTriangle[unit=0.23]
15 \providecommand{\PstPolygonNode}{%
16   \rput{*0}(1;\INode){\small\the\multidocount}}
17 \PstHexagon[unit=0.8]
18 \PstDodecagon[unit=1.2]
19 \PstDodecagon[unit=1.6]
20 \end{pspicture}

```



## References

- [1] Denis Girou, Présentation de PSTricks, *Cahiers GUTenberg*, Number 16, pages 21–70, February 1994 (in french).
- [2] Denis Girou, Building high level objects in PSTricks, TUG’95, St Petersburg, Florida, 1995. Available at [http://www.tug.org/applications/PSTricks/TUG95-PSTricks\\_4.ps.gz](http://www.tug.org/applications/PSTricks/TUG95-PSTricks_4.ps.gz)
- [3] Michel Goossens, Sebastian Rahtz and Frank Mittelbach, *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion*, Addison-Wesley, 1997.
- [4] Alan Hoenig, *T<sub>E</sub>X Unbound: L<sup>A</sup>T<sub>E</sub>X & T<sub>E</sub>X Strategies, Fonts, Graphics, and More*, Oxford University Press, 1998.
- [5] Kristoffer H. Rose and Ross Moore, X<sub>y</sub>-pic. Polygon feature, available from CTAN, 1991-1998, `macros/generic/diagrams/xypic`.
- [6] Timothy van Zandt, PSTricks. PostScript macros for Generic T<sub>E</sub>X, available from CTAN, 1993, `graphics/pstricks`.
- [7] Timothy van Zandt and Denis Girou, Inside PSTricks, *TUGboat*, Volume 15, Number 3, pages 239–246, September 1994.