‘infix-RPN’ – ‘pst-infixplot’

ver. 0.11

Jean-Côme Charpentier <jean-côme.charpentier@wanadoo.fr>
Christophe Jorssen <christophe.jorssen@libre.fr.invalid>
‘libre’ is the french word for ‘free’

2005/05/16

Abstract

Plotting functions with `pst-plot` is very powerful but sometimes difficult to learn since the syntax of `\psplot` and `\parametricplot` requires some PostScript knowledge. What **‘infix-RPN’** and **‘pst-infixplot’** intend to do is to simplify the usage of `pst-plot` for the beginner, providing macro commands that convert natural mathematical expressions to PS syntax.

1 Basic examples: usage of ‘infix-RPN’

`\usepackage{infix-RPN}` for LaTeX users or `\input infix-RPN.tex` for TeX users gives access to three macros: `\infixtoRPN`, `\RPN` and `\DeclareNewPSOperator`.

The macro `\infixtoRPN` takes an infix expression as argument and converts it to Reverse Polish Notation. The result of the conversion is put in the macro `\RPN`.

2 3 add 4 x mul sub
x neg log
2 x y div sin mul

```
\infixtoRPN{2+3-4*x}\RPN

\infixtoRPN{log(-x)}\RPN

\infixtoRPN{2*sin(x/y)}\RPN
```

Multiple signs are OK:

3 0.5 neg neg neg add

```
\infixtoRPN{3+-+--0.5}\RPN
```

For operators that require more than one argument, arguments must be separated with commas:

x y atan

```
\infixtoRPN{atan(x,y)}\RPN
```

There is a difference between variables and operators. There are 11 pre-defined operators[1,2] which are basically those of PostScript: `abs`, `sin`, `cos`, `atan`, `neg`, `ceiling`, `floor`, `truncate`, `sqrt`, `ln`, `log`. You can define more operators with the `\DeclareNewPSOperator` macro:

Div is a PS operator defined by pstricks

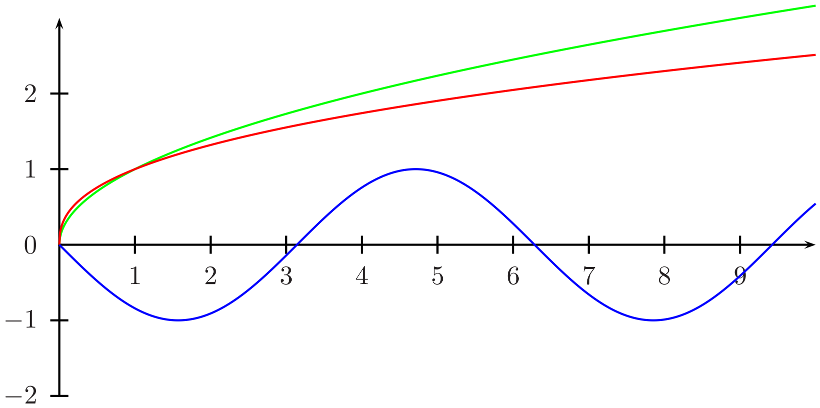
x y Div

```
\DeclareNewPSOperator{Div}
\infixtoRPN{Div(x,y)}\RPN
```

[1] Actually, there are five more operators defined : `add`, `sub`, `mul`, `div` and `exp`. Those ones should *not* be used direclty. Use `+`, `-`, `*`, `/` and `^` instead, which is, by the way, the main interest of using infix notation.

[2] If you use `pst-math` with `infix-RPN`, PS operators added by `pst-math` are declared by `\DeclareNewPSOperator` and are therefore directly accessible in any infix expression.

2 Plot examples with ‘infix-RPN’



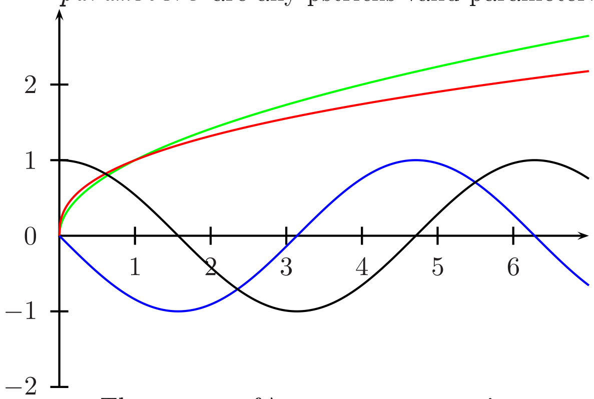

```
\psset{plotpoints=1000}
\psaxes{->}(0,0)(0,-2)(10,3)
\infixtoRPN{sqrt(x)}
\psplot[linecolor=green]{0}{10}{\RPN}
\infixtoRPN{x^0.4}
\psplot[linecolor=red]{0}{10}{\RPN}
\infixtoRPN{sin(-x*180/3.1415)}
\psplot[linecolor=blue]{0}{10}{\RPN}
```

3 Plot examples with ‘pst-infixplot’

If you don’t want the limitation of having to invoke two macro calls (namely `\infixtoRPN` and `\RPN`) for plotting, then use the **‘pst-infixplot’** package! LaTeX users should type `\usepackage{pst-infixplot}` in the preamble when TeX users should type `\input pst-infixplot.tex`.

‘pst-infixplot’ automatically loads `pstricks`, `pst-plot` and `infix-RPN`. **‘pst-infixplot’** defines two macro commands: `\psPlot` and `\parametricPlot`.

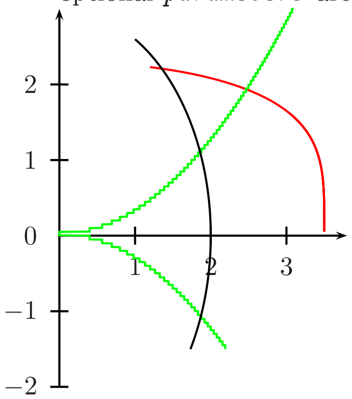
The syntax of `\psPlot` is: `\psPlot[<parameters>]{x_begin}{x_end}{infix_expression}` where the optional *parameters* are any pstricks valid parameter.




```
\psset{plotpoints=1000}
\psaxes{->}(0,0)(0,-2)(7,3)
\psPlot[linecolor=green]{0}{7}{sqrt(x)}
\psPlot[linecolor=red]{0}{7}{x^0.4}
\psPlot[linecolor=blue]{0}{7}{sin(-x*180/3.1415)}
\psplot{0}{7}{x neg 180 mul 3.1415 div cos}
```

The syntax of `\parametricPlot` is:

`\parametricPlot[<parameters>]{x_begin}{x_end}{infix_x_expression}{infix_y_expression}` where the optional *parameters* are any pstricks valid parameter.




```
\psset{plotpoints=1000}
\psaxes{->}(0,0)(0,-2)(3.8,3)
\parametricPlot[linecolor=red]{-30}{70}{3.5*cos(t)}{2.3*sqrt(abs(sin(t)))}
\parametricPlot[linecolor=green]{-30}{60}{4*sqrt(abs(floor(t)))/10}{t/20}
\parametricplot{-30}{60}{2 t cos mul 3 t sin mul}
```