PSTricks

pst-math

Special mathematical PostScript functions; v.0.64

December 15, 2018

Documentation by
Herbert Voß

Package author(s):
Christoph Jorssen
Herbert Voß

Contents

Abstract

`pst-math` is an extension to the PostScript language. The files `pst-math.sty` and `pst-math.tex` are only wrapper files for the `pst-math.pro` file, which defines all the new mathematical functions for use with PostScript.

Thanks to:
Denis Bitouzé; Jacques L'helgoualc'h; Patrice Mégret; Dominik Rodriguez

1 Introduction

`pst-math` defines `\pstPi` on TeX level which expects 1,2,3 or 4 as parameter. It is not available on PostScript level.

$\boxed{\texttt{\textbackslash pstPI\#}}$

`\pstPI1` $\Rightarrow \pi$
`\pstPI2` $\Rightarrow \frac{\pi}{2}$
`\pstPI3` $\Rightarrow \frac{\pi}{3}$
`\pstPI4` $\Rightarrow \frac{\pi}{4}$

2 Trigonometry

`pst-math` introduces natural trigonometric PostScript operators `COS`, `SIN` and `TAN` defined by

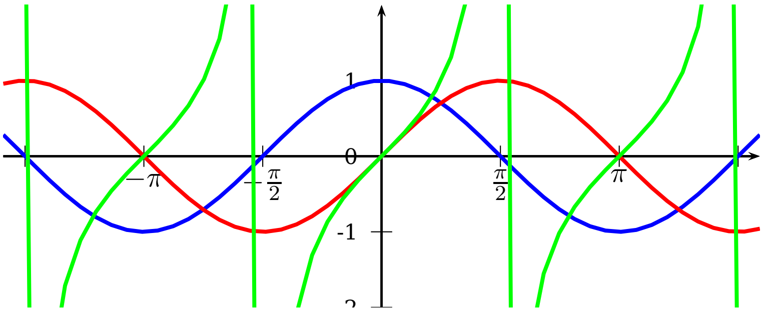
$$\cos : \left\{ \begin{array}{ccc} \mathbb{R} & \rightarrow & [-1,1] \\ x & \mapsto & \cos(x) \end{array} \right.$$

$$\sin : \left\{ \begin{array}{ccc} \mathbb{R} & \rightarrow & [-1,1] \\ x & \mapsto & \sin(x) \end{array} \right.$$

$$\tan : \left\{ \begin{array}{ccc} \mathbb{R}\backslash\{k\frac{\pi}{2}, k \in \mathbb{Z}\} & \rightarrow & \mathbb{R} \\ x & \mapsto & \tan(x) \end{array} \right.$$

where x is in *radians*. `TAN` does *not* produce a PS error[1] when $x = k\frac{\pi}{2}$.

Stack	*Operator*	*Result*	*Description*
num	`COS`	real	Return cosine of num radians
num	`SIN`	real	Return sine of num radians
num	`TAN`	real	Return tangent of num radians



```
\begin{pspicture}*(-5,-2)(5,2)
\SpecialCoor % For label positionning
\psaxes[labels=y,Dx=\pstPI2]{->}(0,0)(-5,-2)(5,2)
\uput[-90](!PI 0){$\pi$} \uput[-90](!PI neg 0){$-\pi$}
\uput[-90](!PI 2 div 0){$\frac{\pi}2$}
```

[1] TAN is defined with Div, a special PSTricks operator rather than with div, the default PS operator.

```
\uput[-90](!PI 2 div neg 0){$-\frac{\pi}2$}
\psplot[linewidth=1.5pt,linecolor=blue]{-5}{5}{x COS}
\psplot[linewidth=1.5pt,linecolor=red]{-5}{5}{x SIN}
\psplot[linewidth=1.5pt,linecolor=green]{-5}{5}{x TAN}
\end{pspicture}
```

pst-math introduces natural trigonometric postscript operators ACOS, ASIN and ATAN defined by

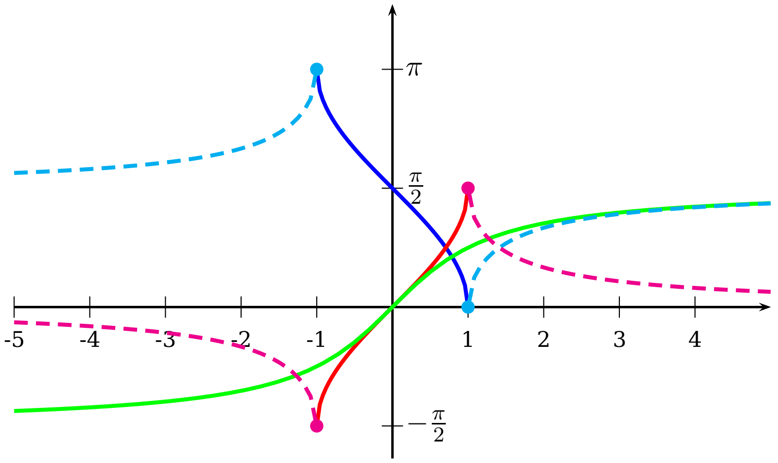
$$\mathrm{acos}:\left\{\begin{array}{ccc}[-1,1] & \to & [0,\pi]\\ x & \mapsto & \mathrm{acos}(x)\end{array}\right.$$

$$\mathrm{asin}:\left\{\begin{array}{ccc}[-1,1] & \to & [-\frac{\pi}{2},\frac{\pi}{2}]\\ x & \mapsto & \mathrm{asin}(x)\end{array}\right.$$

$$\mathrm{atan}:\left\{\begin{array}{ccc}\mathbb{R} & \to &]-\frac{\pi}{2},\frac{\pi}{2}[\\ x & \mapsto & \mathrm{atan}(x)\end{array}\right.$$

Stack	*Operator*	*Result*	*Description*
num	ACOS	angle	Return arccosine of num in radians
num	ASIN	angle	Return arcsine of num in radians
num	ATAN	angle	Return arctangent of num in radians
num	ASEC	angle	Return arcsecans of num in radians
num	ACSC	angle	Return arccosecans of num in radians

ATAN is *not* defined as the already existing PS operator atan. ATAN needs only *one* argument on the stack.



```
\begin{pspicture}(-5,-2)(5,4)
\SpecialCoor % For label positionning
\psaxes[labels=x,Dy=\pstPI2]{->}(0,0)(-5,-2)(5,4)
\uput[0](!0 PI){$\pi$} \uput[0](!0 PI 2 div){$\frac{\pi}2$}
\uput[0](!0 PI 2 div neg){$-\frac{\pi}2$}
\psset{linewidth=1.5pt,yMaxValue=4}
\psplot[linecolor=blue]{-1}{1}{x ACOS} \psplot[linecolor=red]{-1}{1}{x ASIN}
\psplot[linecolor=green]{-5}{5}{x ATAN}
\psplot[linestyle=dashed,linecolor=cyan,arrows=-*]{-5}{-1}{x ASEC}
\psplot[linestyle=dashed,linecolor=cyan,arrows=*-]{1}{5}{x ASEC}
\psplot[linestyle=dashed,linecolor=magenta,arrows=-*]{-5}{-1}{x ACSC}
\psplot[linestyle=dashed,linecolor=magenta,arrows=*-]{1}{5}{x ACSC}
\end{pspicture}
```

3 Hyperbolic trigonometry

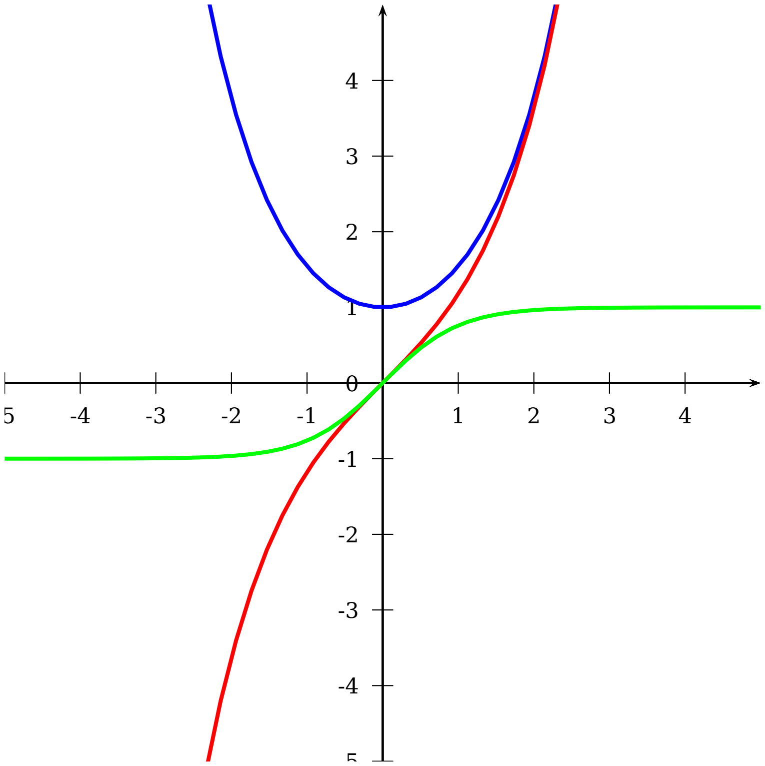
`pst-math` introduces hyperbolic trigonometric postscript operators `COSH`, `SINH` and `TANH` defined by

$$\cosh : \left\{ \begin{array}{ccc} \mathbb{R} & \to & [1, +\infty[\\ x & \mapsto & \cosh(x) \end{array} \right.$$

$$\sinh : \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \sinh(x) \end{array} \right.$$

$$\tanh : \left\{ \begin{array}{ccc} \mathbb{R} & \to &]-1, 1[\\ x & \mapsto & \tanh(x) \end{array} \right.$$

Stack	*Operator*	*Result*	*Description*
num	`COSH`	real	Return hyperbolic cosine of num
num	`SINH`	real	Return hyperbolic sine of num
num	`TANH`	real	Return hyperbolic tangent of num



```
\begin{pspicture}*(-5,-5)(5,5)
\psaxes{->}(0,0)(-5,-5)(5,5)
\psplot[linewidth=1.5pt,linecolor=blue]{-5}{5}{x COSH}
\psplot[linewidth=1.5pt,linecolor=red]{-5}{5}{x SINH}
\psplot[linewidth=1.5pt,linecolor=green]{-5}{5}{x TANH}
\end{pspicture}
```

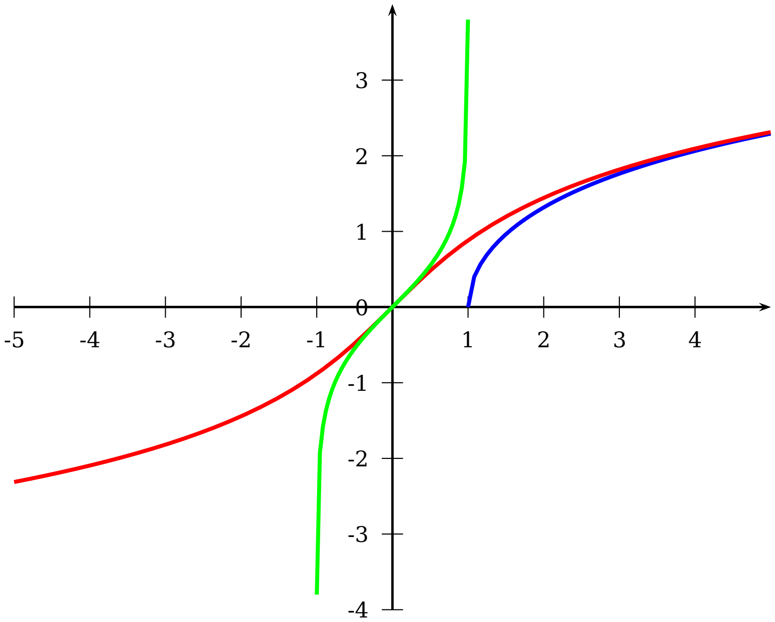
`pst-math` introduces reciprocal hyperbolic trigonometric postscript operators `ACOSH`, `ASINH` and `ATANH` defined by

$$\operatorname{acosh} : \left\{ \begin{array}{ccc} [1, +\infty[& \to & \mathbb{R} \\ x & \mapsto & \operatorname{acosh}(x) \end{array} \right.$$

$$\operatorname{asinh} : \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \operatorname{asinh}(x) \end{array} \right.$$

$$\operatorname{atanh}: \left\{ \begin{array}{ccc}]-1,1[& \to & \mathbb{R} \\ x & \mapsto & \operatorname{atanh}(x) \end{array} \right.$$

Stack	*Operator*	*Result*	*Description*
num	ACOSH	real	Return reciprocal hyperbolic cosine of num
num	ASINH	real	Return reciprocal hyperbolic sine of num
num	ATANH	real	Return reciprocal hyperbolic tangent of num



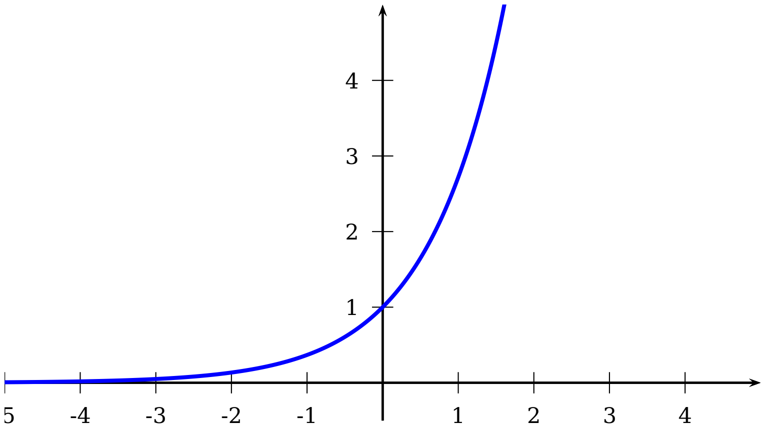
```
\begin{pspicture}(-5,-4)(5,4)
\psaxes{->}(0,0)(-5,-4)(5,4)
\psplot[linewidth=1.5pt,linecolor=blue]{1}{5}{x ACOSH}
\psplot[linewidth=1.5pt,linecolor=red]{-5}{5}{x ASINH}
\psplot[linewidth=1.5pt,linecolor=green]{-.999}{.999}{x ATANH}
\end{pspicture}
```

4 Other operators

`pst-math` introduces postscript operator EXP defined by

$$\exp: \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \exp(x) \end{array} \right.$$

Stack	*Operator*	*Result*	*Description*
num	EXP	real	Return exponential of num

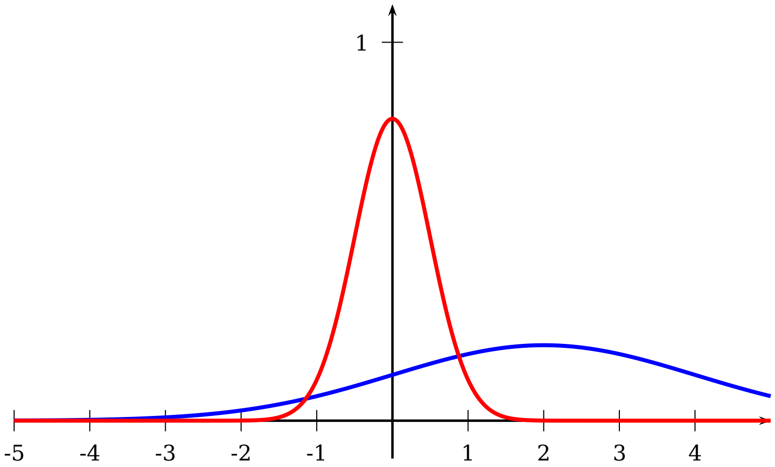


```
\begin{pspicture}*(-5,-1)(5,5)
\psaxes{->}(0,0)(-5,-0.5)(5,5)
\psplot[linecolor=blue,linewidth=1.5pt,plotpoints=1000]{-5}{5}{x EXP}
\end{pspicture}
```

`pst-math` introduces postscript operator GAUSS defined by

$$\text{gauss} : \left\{\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \dfrac{1}{\sqrt{2\pi\sigma^2}} \exp - \dfrac{(x-\overline{x})^2}{2\sigma^2} \end{array}\right.$$

Stack	*Operator*	*Result*	*Description*
num1 num2 num3	GAUSS	real	Return gaussian of num1 with mean num2 and standard deviation num3

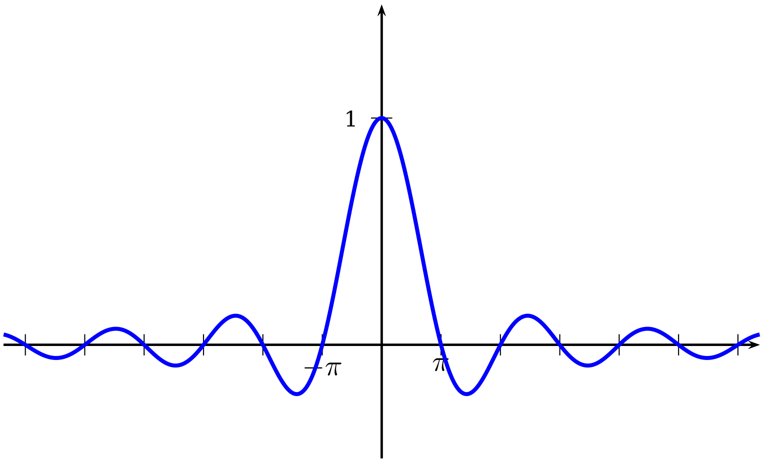


```
\psset{yunit=5}
\begin{pspicture}(-5,-.1)(5,1.1)
\psaxes{->}(0,0)(-5,-.1)(5,1.1)
\psplot[linecolor=blue,linewidth=1.5pt,plotpoints=1000]{-5}{5}{x 2 2 GAUSS}
\psplot[linecolor=red,linewidth=1.5pt,plotpoints=1000]{-5}{5}{x 0 .5 GAUSS}
\end{pspicture}
```

`pst-math` introduces postscript operator SINC defined by

$$\text{sinc} : \left\{\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \dfrac{\sin x}{x} \end{array}\right.$$

Stack	*Operator*	*Result*	*Description*
num	SINC	real	Return cardinal sine of num radians



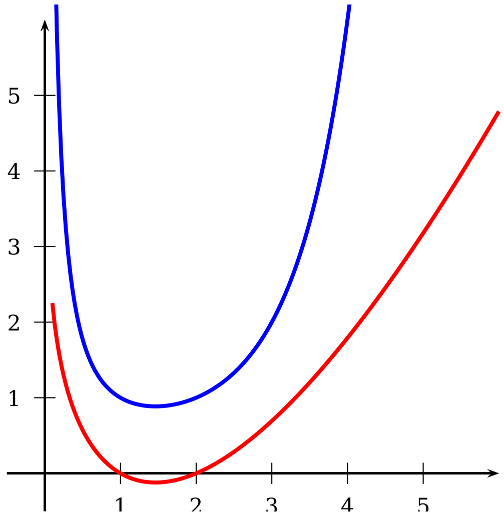
```
\psset{xunit=.25,yunit=3}
\begin{pspicture}(-20,-.5)(20,1.5)
\SpecialCoor % For label positionning
\psaxes[labels=y,Dx=\pstPI1]{->}(0,0)(-20,-.5)(20,1.5)
\uput[-90](!PI 0){$\pi$} \uput[-90](!PI neg 0){$-\pi$}
\psplot[linecolor=blue,linewidth=1.5pt,plotpoints=1000]{-20}{20}{x SINC}
\end{pspicture}
```

`pst-math` introduces postscript operator `GAMMA` and `GAMMALN` defined by

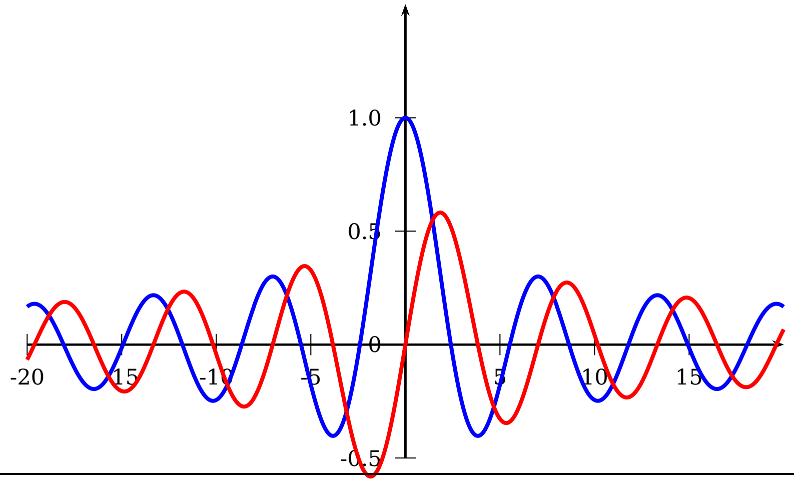
$$\Gamma : \left\{ \begin{array}{ccc} \mathbb{R}\backslash\mathbb{Z} & \to & \mathbb{R} \\ x & \mapsto & \displaystyle\int_0^{\infty} t^{x-1}\mathrm{e}^{-t}\,\mathrm{d}t \end{array} \right.$$

$$\ln\Gamma : \left\{ \begin{array}{ccc}]0,+\infty[& \to & \mathbb{R} \\ x & \mapsto & \ln\displaystyle\int_0^{t} t^{x-1}\mathrm{e}^{-t}\,\mathrm{d}t \end{array} \right.$$

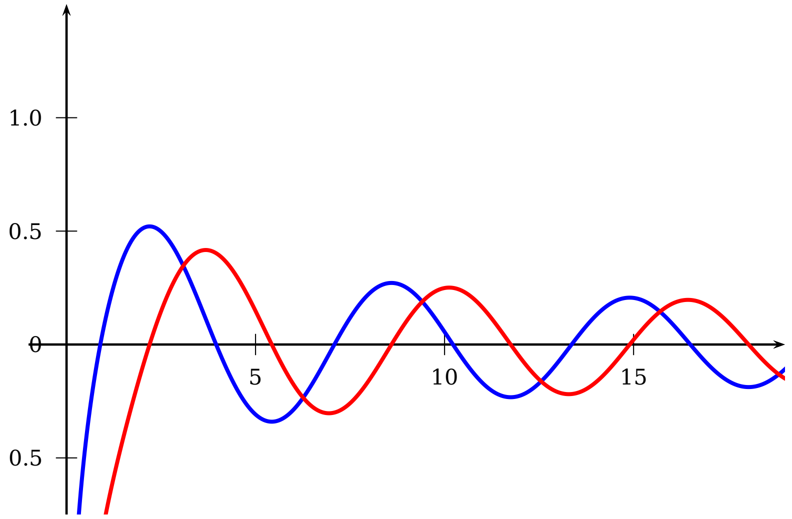
Stack	*Operator*	*Result*	*Description*
num	GAMMA	real	Return Γ function of num
num	GAMMALN	real	Return logarithm of Γ function of num



```
\begin{pspicture*}(-.5,-.5)(6.2,6.2)
\psaxes{->}(0,0)(-.5,-.5)(6,6)
\psplot[linecolor=blue,linewidth=1.5pt,plotpoints=200]{.1}{6}{x GAMMA}
\psplot[linecolor=red,linewidth=1.5pt,plotpoints=200]{.1}{6}{x GAMMALN}
\end{pspicture*}
```



```
\psset{xunit=.25,yunit=3}
\begin{pspicture}(-20,-.5)(20,1.5)
\psaxes[Dx=5,Dy=.5]{->}(0,0)(-20,-.5)(20,1.5)
\psplot[linecolor=blue,linewidth=1.5pt,plotpoints=1000]{-20}{20}{x BESSEL_J0}
\psplot[linecolor=red,linewidth=1.5pt,plotpoints=1000]{-20}{20}{x BESSEL_J1}
\end{pspicture}
```

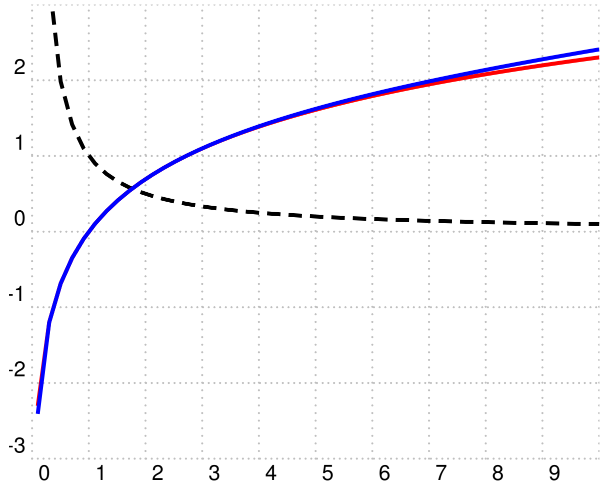


```
\psset{xunit=.5,yunit=3}
\begin{pspicture}*(-1.5,-.75)(19,1.5)
\psaxes[Dx=5,Dy=.5]{->}(0,0)(-1,-.75)(19,1.5)
\psplot[linecolor=blue,linewidth=1.5pt,plotpoints=1000]{0.0001}{20}{x BESSEL_Y0}
\psplot[linecolor=red,linewidth=1.5pt,plotpoints=1000]{0.0001}{20}{x BESSEL_Y1}
%\psplot[linecolor=green,plotpoints=1000]{0.0001}{20}{x 2 BESSEL_Yn}
\end{pspicture}
```

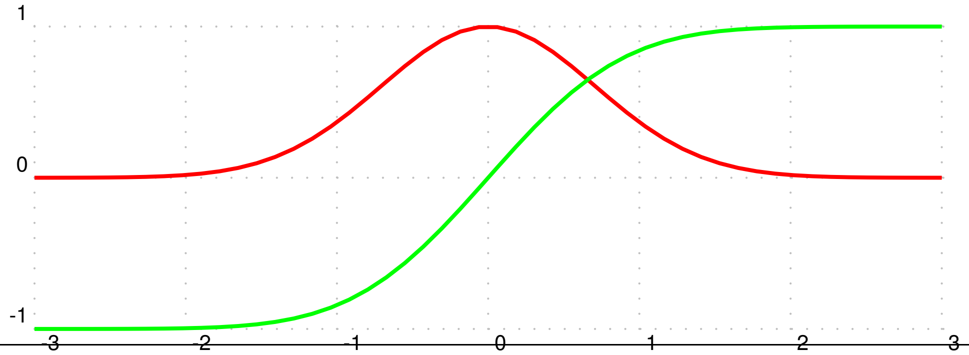
5 Numerical integration

Stack	*Operator*	*Result*	*Description*
num num /var { function } num num	`SIMPSON`	real	Return $\int\limits_a^b f(t)\mathrm{d}t$

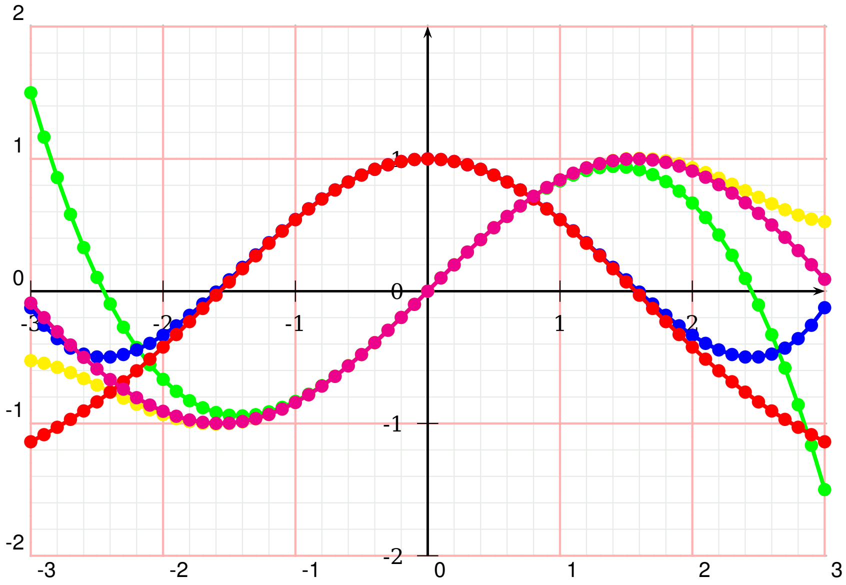
the first two variables are the low and high boundary integral, both can be values or `PostScript` expressions. `/var` is the definition of the integrated variable (not x!), which is used in the following function description, which must be inside of braces. The last number is the tolerance for the step adjustment. The function `SIMPSON` can be nested.



```
\psset{xunit=.75}
\begin{pspicture*}[showgrid=true](-0.4,-3.4)(10,3)
  \psplot[linestyle=dashed,linewidth=1.5pt]{.1}{10}{1 x div}
  \psplot[linecolor=red,linewidth=1.5pt]{.1}{10}{
    1         % start
    x         % end
    /t        % variable
    { 1 t div } % function
    .001      % tolerance
    SIMPSON } %
  \psplot[linecolor=blue,linewidth=1.5pt]{.1}{10}{1 x /t { 1 t div } 1 SIMPSON }
\end{pspicture*}
```



```
%%% Gaussian and relative integral from -x to x to its value sqrt{pi}
\psset{unit=2}
\begin{pspicture}[showgrid=true](-3,-1)(3,1)
  \psplot[linecolor=red,linewidth=1.5pt]{-3}{3}{Euler x dup mul neg exp }
  \psplot[linecolor=green,linewidth=1.5pt]{-3}{3}
    { x neg x /t { Euler t dup mul neg exp } .001 SIMPSON Pi sqrt div}
\end{pspicture}
```



```
\psset{unit=1.75cm}
%%% successive polynomial developments of sine-cosine
\begin{pspicture}[showgrid=true](-3,-2)(3,2)
\psaxes{->}(0,0)(-3,-2)(3,2)
 \psplot[linecolor=green, algebraic=false, plotpoints=61, showpoints=true]
      {-3}{3}{0 x /tutu
            {1 0 tutu /toto { toto } .1 SIMPSON sub}
              .01 SIMPSON }
 \psplot[linecolor=blue, algebraic=false, plotpoints=61, showpoints=true]
      {-3}{3}{1 0 x /tata
            {0 tata /tutu
             {1 0 tutu /toto { toto } .1 SIMPSON sub}
              .01 SIMPSON }
               .01 SIMPSON sub}
 \psplot[linecolor=yellow, algebraic=false, plotpoints=61, showpoints=true]
      {-3}{3}{0 x /titi
            {1 0 titi /tata
             {0 tata /tutu
              {1 0 tutu /toto { toto } .1 SIMPSON sub}
               .01 SIMPSON }
                .01 SIMPSON sub}
                 .01 SIMPSON }
 \psplot[linecolor=red, algebraic=false, plotpoints=61, showpoints=true]
      {-3}{3}{1 0 x /tyty
            {0 tyty /titi
             {1 0 titi /tata
              {0 tata /tutu
               {1 0 tutu /toto { toto } .1 SIMPSON sub}
                .01 SIMPSON }
                 .01 SIMPSON sub}
                  .01 SIMPSON }
                   .01 SIMPSON sub}
 \psplot[linecolor=magenta, algebraic=false, plotpoints=61, showpoints=true]
      {-3}{3}{0 x /tete
            {1 0 tete /tyty
             {0 tyty /titi
              {1 0 titi /tata
               {0 tata /tutu
                {1 0 tutu /toto { toto } .1 SIMPSON sub}
                 .01 SIMPSON }
```

```
                    .01 SIMPSON sub}
                     .01 SIMPSON }
                      .01 SIMPSON sub}
                       .01 SIMPSON }%%% FIVE nested calls
\end{pspicture}
```

6 Random numbers

Package `pst-math` supports the creation of random number lists where a number will appear only once.

```
\defineRandIntervall(min,max)maxNo
\makeSimpleRandomNumberList% multiple values possible
\makeRandomNumberList % no multiple values!
\getNumberFromList{number}
```

The list of the random numbers is `\RandomNumbers`, a comma separated list of the values. It can be used for own purpuses.

Random list: 39,19,6,11,2,10,6,10,1,32,50,4,28,45,19,40,5,2,3,27,3,45,12,47,12,13,44,20,16,22
39 19 6 11 2 10 6 10 1 32 50 4 28 45 19 40 5 2 3 27 3 45 12 47 12 13 44 20 16 22

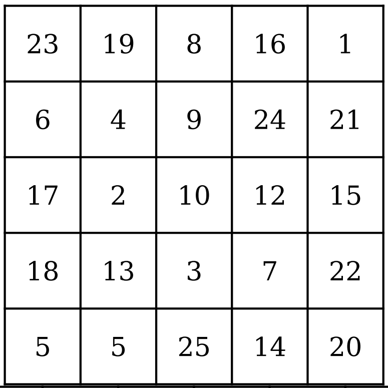
```
\defineRandIntervall(1,50){30}
\makeSimpleRandomNumberList
Random list: \RandomNumbers

\psforeach{\iA}{1,2,..,30}{\getNumberFromList{\iA}~}
```

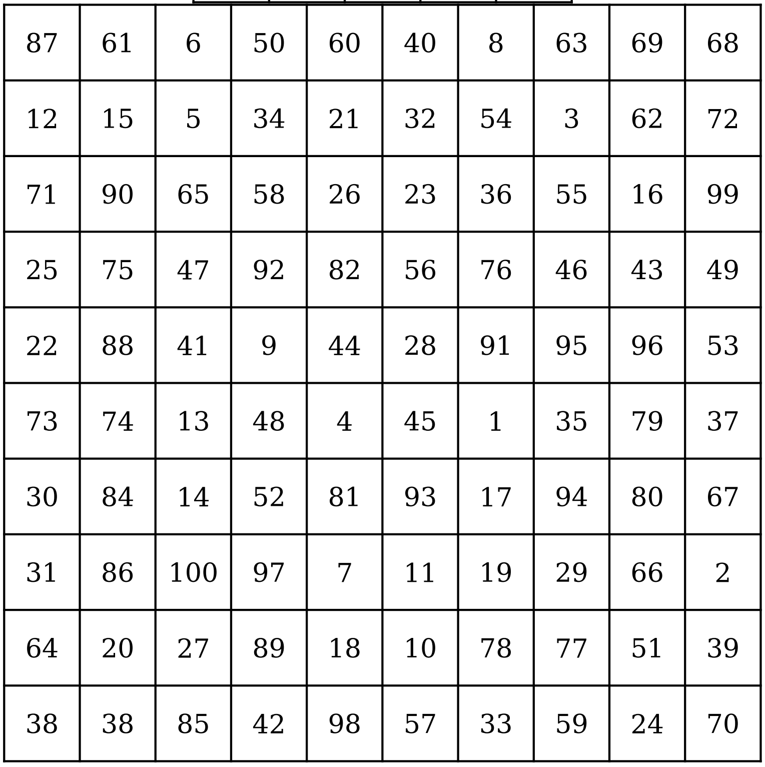
In the next example a random number appears only *once* in the list. There are no multiple numbers:

13 13 25 6 2 19 4 7 16 8 3 11 21 17 27 1 29 22 9 12 30 24 26 5 18 20 23 14 10 15

```
\defineRandIntervall(1,30){30}
\makeRandomNumberList
\psforeach{\iA}{1,2,..,30}{\getNumberFromList{\iA}~}
```



23	19	8	16	1
6	4	9	24	21
17	2	10	12	15
18	13	3	7	22
5	5	25	14	20

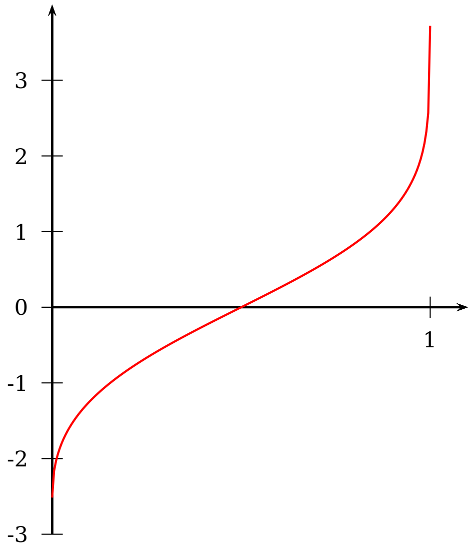


87	61	6	50	60	40	8	63	69	68
12	15	5	34	21	32	54	3	62	72
71	90	65	58	26	23	36	55	16	99
25	75	47	92	82	56	76	46	43	49
22	88	41	9	44	28	91	95	96	53
73	74	13	48	4	45	1	35	79	37
30	84	14	52	81	93	17	94	80	67
31	86	100	97	7	11	19	29	66	2
64	20	27	89	18	10	78	77	51	39
38	38	85	42	98	57	33	59	24	70

```
\newcounter{RandNo}
\def\n{5} \def\N{\the\numexpr\n*\n}
\defineRandIntervall(1,\N){\N}
\makeRandomNumberList \setcounter{RandNo}{1}
\begin{pspicture}(\n,\n)
  \psgrid[subgriddiv=0,gridlabels=0pt]
  \multido{\rRow=0.5+1.0}{\n}{\multido{\rCol=0.5+1.0}{\n}{%
      \rput(\rCol,\rRow){\getNumberFromList{\theRandNo}}%
     \stepcounter{RandNo}}}
\end{pspicture}
\setcounter{RandNo}{1}
\def\n{10} \def\N{\the\numexpr\n*\n}
\defineRandIntervall(1,\N){\N}
\makeRandomNumberList \setcounter{RandNo}{1}
\begin{pspicture}(\n,\n)
  \psgrid[subgriddiv=0,gridlabels=0pt]
  \multido{\rRow=0.5+1.0}{\n}{\multido{\rCol=0.5+1.0}{\n}{%
      \rput(\rCol,\rRow){\getNumberFromList{\theRandNo}}%
     \stepcounter{RandNo}}}
\end{pspicture}
```

7 Numerical functions

Stack	*Operator*	*Result*	*Description*
num	norminv	real	Return $norminv(num)$



```
\psset{xunit=5}
\begin{pspicture}(-0.1,-3)(1.1,4)
\psaxes{->}(0,0)(0,-3)(1.1,4)
\psplot[plotpoints=200,algebraic,linecolor=red]{0}{0.9999}{norminv(x)}
\end{pspicture}
```

These function returns the inverse normal.

References

[1] Denis Girou. “Présentation de PSTricks”. In: *Cahier GUTenberg* 16 (Apr. 1994), pp. 21–70.

[2] Michel Goosens et al. *The LATEX Graphics Companion*. Reading, Mass.: Addison-Wesley Publishing Company, 2007.

[3] Laura E. Jackson and Herbert Voß. “Die Plot-Funktionen von `pst-plot`”. In: *Die TEXnische Komödie* 2/02 (June 2002), pp. 27–34.

[4] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. Vaterstetten: IWT, 1989.

[5] Herbert Voß. “Die mathematischen Funktionen von PostScript”. In: *Die TEXnische Komödie* 1/02 (Mar. 2002).

[6] Herbert Voß. *Mathematiksatz mit LATEX*. Heidelberg/Berlin: DANTE/Lehmanns Media, 2009.

[7] Herbert Voß. *`PSTricks` – Grafik für TEX und LATEX*. 5. Heidelberg/Hamburg: DANTE – Lehmanns, 2008.

[8] Eric Weisstein. *Wolfram MathWorld*. `http://mathworld.wolfram.com`: Wolfram, 2007.

[9] Timothy van Zandt. *`multido.tex` - a loop macro, that supports fixed-point addition*. `CTAN:/graphics/pstricks/generic/multido.tex`, 1997.

[10] Timothy van Zandt. *`pst-plot`: Plotting two dimensional functions and data*. `CTAN:graphics/pstricks/generic/pst-plot.tex`, 1999.

[11] Timothy van Zandt. *PSTricks - `PostScript` macros for generic TEX*. `http://www.tug.org/application/PSTricks`, 1993.

[12] Timothy van Zandt and Denis Girou. “Inside PSTricks”. In: *TUGboat* 15 (Sept. 1994), pp. 239–246.

Index