l3draw Learn

Eureka

June 2, 2024

Contents

1 Basic

1.1 unit

Default unit of l3draw is **pt**.

1.2 How to displace in Doc

Except for function `\draw_path_use_clear:n {}`, all the other functions don't display anything on screen.

1.3 polar point

$$(\rho, \theta) \leftrightarrow (\rho\cos\theta, \rho\sin\theta) \tag{1}$$
$$(\rho_1, \rho_2, \theta) \leftrightarrow (\rho_1\cos\theta, \rho_2\sin\theta) \tag{2}$$

1.4 base vectors

The `\draw_x(y)vec:n` defines 2 base vectors that can be used in command `\draw_point_vec:nn`. So the `\draw_xvec:n` returns nothing, but the `\draw_point_vec:nn` returns a coodinate. The standard base is **1cm** in $x(y)$ direction.

All **vec**-based command is **relative** to the above defined $x(y)-$vectors. That's if you changed the base, the norm of `\draw_point_vec_polar:nn {1}{0}`$= \|e_x\|$, while that of `\draw_point_polar:nn {1}{0}` $\equiv$ 1pt.

1.5 root

The `{root}` in commands related to intersection is a **integer**, reprensent the index of the intersection(s).

1.6 interpolation

Return a point the `<part>`-way of the line between the two points `<point1>` and `<point2>`. Then `<part>` would be a float value.

Or you can use distance instead of portion. The distance is the **absolutely** value of the distance to the **first** point(point-1). see below figure 1:

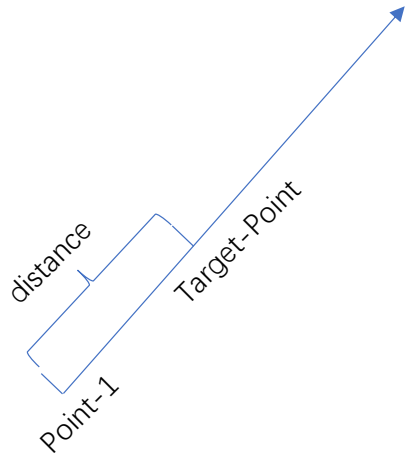

Figure 1: distance

2 Path

arc length = (corner) to (the point curve start). **leading in**:from the previous path operation. so does the **leading out**.

Make a path close, use command `\draw_path_close:`, contrast to `--cycle` in tikz.

2.1 show path

To show a path you can use command `\draw_path_use:n`, using the following action:

- stroke(draw)
- fill
- clip

You can also clear the path after show it, using command `\draw_path_use_clear:n` instead.

2.2 replace bounding box

Replaces the current bounding box of the drawing with one specified by the current path, which means this bb **won't update in the future**. For that "All functions automatically update the bounding box of the image", unless specified `\l_draw_bb_update_bool=false`.

2.3 canva

canvas-related functions ignore the transformation matrix.

2.4 Phase in Path pattern

What is the "phase" in `\draw_dash_pattern:nn` ? Remeber that **left:+, right:-**.

The ⟨phase⟩ specifies (that) the pattern should start where during the first 'on' line
The ⟨phase⟩ specifies where (during the first "on" line) the pattern should start.

2.5 fill rule

there are 2 rules to check whether a region is "internal" of a path:

- **Non-zero winding number rule**
- **Even-odd rule**

3 box in drawing

box in l3draw, like 'node' in tikz.It takes account of the current transformation matrix and shift. But you can use 'xcoffins' to re-set it to the original box.

4 transform matrix

`\draw_transform_matrix:nnnn {a}{b}{c}{d}` corresponding to a matrix(denoted by T) below:

$$T = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \tag{3}$$

that's "arranged column by column". You can **shift, scale, rotate, slant** the drawing using pre-defined functions. But all the functions are juse special cases to the previous transformation matrix.

4.1 invert and reset

invert applys the inverse of current transforming matrix to the drawing, that is applying the following matrix:

$$T = \frac{1}{ad - bc} \begin{bmatrix} d & -c \\ -b & a \end{bmatrix} \tag{4}$$

Whilst, **reset** means: reset the current transforming matrix to the identity matrix, that is

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

5 scope and layers

layers is important in drawing. Default layer is `main`. Create a new layer using `\draw_layer_new:n {newlayer}`, then use it in scope `\draw_layer_begin:n {newlayer} <draw code>\draw_layer_end:`.

Note that each scope or layer loacl environment should have a `\draw_path_use:n {}` function to show the scope drawing material.

6 Draw style

6.1 draw path action

you can **draw(stroke), fill, clip** a path, thus you can specify the color of the path using `\color_fill:n` or `\color_stroke:n`. The stroke color will apply to both **path** and **text** in the drawing.

6.2 opacity

To support "opacity" in l3draw, add `\DocumentMetadata{}` before `\documentclass{}`

7 Implement

7.1 function plot

```
% ===> function plot and shade command
% \function_plot:nnn {<function>}{<domain>}{<style>}
% functin = {<function>}
% domain = {<x-start>, <x-step>, <x-end>, <y-min>, <y-max>}
% style = {<unit>, <action>, <color-1>, <color-2>, <gradient-axis>}
\cs_generate_variant:Nn \fp_step_inline:nnnn {eeen}
\cs_set:Npn \function_plot:nnn #1#2#3 {
  % => split arg
  \args_split_cs:nn {#2}{domain}
  \args_split_cs:nn {#3}{style}
  % => draw function
  \draw_begin:
    % normal part
    \str_case:VnT \l__arg_style_ii {
      {stroke}{\exp_args:Nx \color_stroke:n {\tl_use:N \l__arg_style_iii}}
      {draw}{\exp_args:Nx \color_stroke:n {\tl_use:N \l__arg_style_iii}}
      {fill}{\exp_args:Nx \color_fill:n {\tl_use:N \l__arg_style_iii}}
      {clip}{\relax}
    }{
      % => start point
      \tl_set:Nn \l_tmpa_tl {#1}
      \tl_replace_all:Nne \l_tmpa_tl {x}{(\tl_use:N \l__arg_domain_i)}
      \draw_path_moveto:n {\l__arg_domain_i\l__arg_style_i, \l_tmpa_tl\l__arg_style_i}
      % loop to draw path
      \fp_step_inline:eeen {\l__arg_domain_i}{\l__arg_domain_ii}{\l__arg_domain_iii}{
          \tl_set:Nn \l_tmpa_tl {#1}
          \tl_replace_all:Nnn \l_tmpa_tl {x}{(##1)}
          \draw_path_lineto:n {##1 \l__arg_style_i, \l_tmpa_tl \l__arg_style_i}
      }
      \draw_path_use_clear:n {\l__arg_style_ii}
    }
    % shade plot part
    \str_if_eq:VnT \l__arg_style_ii {shade}{
      % start and end point for 'y-axis gradient'
      \tl_if_eq:VnT \l__arg_style_v {y}{
        \tl_if_exist:cF {l__start_tl}{
          \tl_new:N \l__start_tl
          \tl_new:N \l__end_tl
        }
        \tl_set:Nx \l__start_tl {\l__arg_domain_iv}
        \tl_set:Nx \l__end_tl {\l__arg_domain_v}
      }
      % loop to plot segements
      \fp_step_inline:eeen {\l__arg_domain_i}{\l__arg_domain_ii}{\fp_eval:n {\l__arg_domain_iii-\l__arg_domain_ii}}{
          \tl_set:Nn \l_tmpa_tl {#1}
          \tl_set:Nn \l_tmpb_tl {#1}
          \tl_replace_all:Nnn \l_tmpa_tl {x}{(##1)}
          \tl_replace_all:Nnn \l_tmpb_tl {x}{(##1+\l__arg_domain_ii)}
          \str_case:VnF \l__arg_style_v {
            {x}{\color_gradient:xxx {
                \fp_eval:n {(##1-\l__arg_domain_i)*(100/(\l__arg_domain_iii-\l__arg_domain_i))}
              }{\l__arg_style_iii}{\l__arg_style_iv}}
            {y}{\color_gradient:xxx {
                \fp_eval:n {(\l_tmpa_tl-\l__start_tl)*(100/(\l__end_tl-\l__start_tl))}
              }{\l__arg_style_iii}{\l__arg_style_iv}}
          }{\relax}
          \draw_path_moveto:n {##1 \l__arg_style_i, \l_tmpa_tl \l__arg_style_i}
          \draw_path_lineto:n {(##1+\l__arg_domain_ii) \l__arg_style_i, \l_tmpb_tl \l__arg_style_i}
          % \draw_cap_rectangle:
          \draw_cap_round:
          \draw_path_use_clear:n {draw}
      }
    }
  \draw_end:
}

% args split
\cs_generate_variant:Nn \int_step_inline:nnn {nen}
\cs_generate_variant:Nn \seq_set_split:Nnn {cnn}
\cs_set:Npn \args_split_cs:nn #1#2 {
  \seq_if_exist:cF {l_#2_seq}{
    \seq_new:c {l_#2_seq}
```

```
  }
  \seq_set_split:cnn {l_#2_seq}{,}{#1}
  \int_step_inline:nen {1}{\seq_count:c {l_#2_seq}}{
    \tl_set:Nn \l_tmpa_tl {l__arg_#2_\int_to_roman:n{##1}}
    \exp_args:Nfo \tl_if_exist:cF {\tl_use:N \l_tmpa_tl}{
      \tl_new:c {\tl_use:N \l_tmpa_tl}
    }
    \tl_set:ce {\tl_use:N \l_tmpa_tl}{\seq_item:cn {l_#2_seq}{##1}}
  }
}

% color gradient
\cs_set:Npn \color_gradient:nnn #1#2#3 {
  \fp_compare:nNnTF {#1}>{100}{
    \exp_args:Nx \color_select:n {#2!\fp_eval:n{abs(#1)/(\l__arg_domain_v-\l__arg_domain_iv)}!#3}
  }{
    \fp_compare:nNnTF {#1}<{0}{
      \exp_args:Nx \color_select:n {#2!\fp_eval:n{abs(#1)/(\l__arg_domain_v-\l__arg_domain_iv)}!#3}
    }{\color_select:n {#2!#1!#3}}
  }
}
\cs_generate_variant:Nn \color_gradient:nnn {xxx}

\function_plot:nnn {x^3-2*x}{-5, 0.25, 5, -1, 1}{em, shade, blue, green, x}
% \function_plot:nnn {x^3-2*x}{-3, 1, 3, -1, 1}{em, draw, blue, green, x}
% \function_plot:nnn {sin(x)}{0, 0.02*\c_pi_fp, 2*\c_pi_fp, -1, 1}{em, shade, blue, green, y}
```

7.2 Shade plot

8 l3 Draw Example