

PSTricks

pst-marble v 0.1

A PSTricks package to draw marble-like patterns

September 23, 2018

Package author(s):
Aubrey Jaffer
with the help of
Manuel Luque
Jürgen Gilg

Contents

1	Theory of marbling by Aubrey Jaffer	4
1.1	Introduction	4
2	Techniques	5
2.1	Drop ink	5
2.2	Random drops	6
2.3	Concentric rings	7
3	The command \psMarble	8
4	Rendering	9
5	Colors	10
6	Predefined objects before deformation	11
6.1	/ink	11
6.2	concentric-bands	12
7	Basic deformation actions	13
7.1	/line-deformation	13
7.2	/stroke-deformation	17
7.3	/circle-deformation	19
7.4	/vortex-deformation	21
7.5	/wiggle	23
7.6	/offset	27
8	Combined actions	28
8.1	Bouquets	28
8.2	Fern	29
8.3	Draw contours only	31
8.4	Ripple	33
8.5	Line-Line-Wiggle	34
8.6	Blendmode	35
8.7	Transparency	38
9	List of all optional arguments for pst-marble	40

Aubrey Jaffer finds a similarity between whirlwinds in the great spot of jupiter and those that appear in some marbled papers.

<http://voluntocracy.blogspot.com/2018/08/>

You can see a swirl in a marbled paper on wikipedia:

https://fr.wikipedia.org/wiki/Papier_marbr%C3%A9#/media/File:PaperMarbling003France18.jpg

It is true that in both cases, although at very different scales, the laws of fluid mechanics apply.

Aubrey Jaffer's paper on the physical and mathematical interpretation of the formation of various types of marbling:

<https://arxiv.org/abs/1702.02106>

Aubrey Jaffer and other researchers have done a lot of work to model the marbling phenomena and the results are fine and very close to the reality of the papers made by marbling artists.

Aubrey Jaffer coded many postscript examples, so we came up with the idea of adapting them to PSTricks in the usual form of a package.

This program can redistributed and/or modified under the terms of the LaTeX Project Public License Distributed from CTAN archives in directory `macros/latex/base/lppl.txt`; either version 1.3c of the License, or (at your option) any later version.

1 Theory of marbling by Aubrey Jaffer

<https://www.computer.org/csdl/mags/cg/2012/06/mcg2012060026-abs.html>

<https://www.computer.org/csdl/mags/cg/2017/02/mcg2017020090.html>

<https://arxiv.org/pdf/1702.02106>

1.1 Introduction

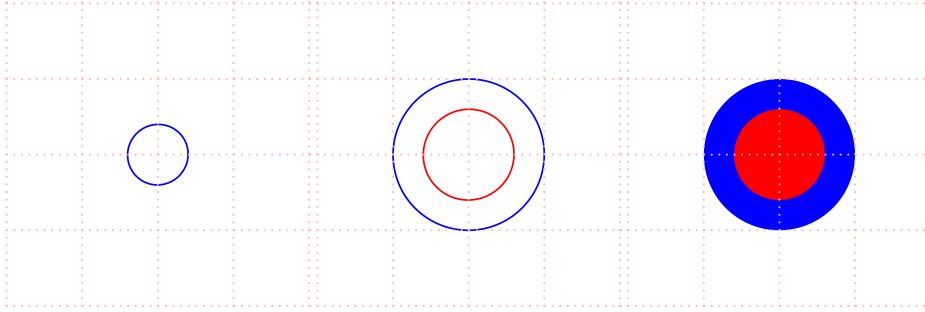
Marbling originated in Asia as a decorative art more than 800 years ago and spread to Europe in the 1500s where it was used for endpapers and book covers.

The mathematical fascination with ink marbling is that while rakings across the tank stretch and deform the ink boundaries, they do not break or change the topology of the surface. With mechanical guides, a raking can be undone by reversing the motion of the rake to its original position. Raking is thus a physical manifestation of a homeomorphism, a continuous function between topological spaces (in this case between a topological space and itself) that has a continuous inverse function.

2 Techniques

2.1 Drop ink

The first drop of ink placed within water forms a circle with the area a . If a second drop with the area b is placed within the center of the first drop, the total area increases from a to $a + b$. For the first drop, points very close to the center will change from an infinitely small radius to a radius $\sqrt{b/\pi}$; and the points of the limit circle will change from $\sqrt{a/\pi}$ to $\sqrt{(a+b)/\pi}$. If we take 2 drops of different colors, this gives:



The command to drop a drop is written as follows:

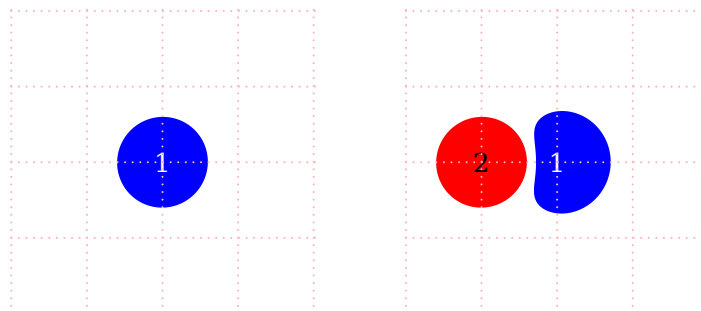
```
xc yc R      r g b
[ 0 0 100 [] [0 0 1] /ink]
```

Note that the coordinates of the center of the drop and its radius are in points¹ and its color is in the system rgb: (values between 0 and 1).

When we place the second drop of radius r at the point $C(x_C, y_C)$, Aubrey Jaffer considers that this one remains round, intact, but that the first then undergoes the influence of the second and deforms according to the law:

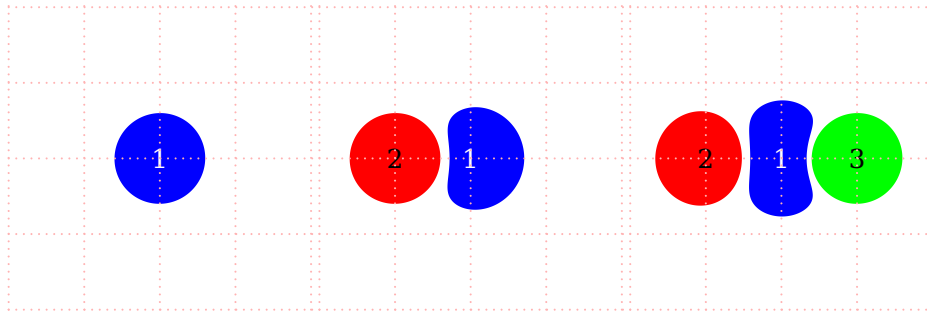
$$\vec{P}' = \vec{C} + \left(\vec{P} - \vec{C} \right) \sqrt{1 + \frac{r^2}{\|\vec{P} - \vec{C}\|^2}}$$

$P(x, y)$ is a point of the first drop and $P'(x', y')$ the transformed point.



If a third drop is placed, the two previous drops will then be influenced by the third, which remains intact.

¹ There is a scaling. Example: if the largest dimension of the page is 4, 100 pts will be represented 0.4 cm



All drops are influenced by the last drop deposited.

2.2 Random drops

One of the techniques is to project with a brush drops of color on the surface of the liquid in several stages by changing color. The position of the drops is therefore random. Each drop influences its neighbors and assuming that initially the drops would form a round spot on the surface, they will deform depending on the size and proximity of neighbors. The modeling of this phenomenon has been studied in the document “*Mathematical Marbling*” by Shufang Lu, Aubrey Jaffer, Xiaogang Jin, Hanli Zhao and Xiaoyang Mao.

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

<https://www.computer.org/csdl/mags/cg/2012/06/mcg2012060026-abs.html>

Then, with a fine stick, a comb one tries to draw the marbling.

The following example illustrates that technique. Three steps with drops of different size and color on which 2 swirls are applied.



```

\begin{pspicture}(-3,-6)(3,6)
\psMarble[L=100,
  actions={
    250 {
      [random:uniform 1000 mul 500 sub % -500 <x<500
        random:uniform 1000 mul 500 sub % -500 <y<500
        30 background [0.960 0.764 0.576] /ink]
    } repeat
    200 {
      [random:uniform 1000 mul 500 sub % -500 <x<500
        random:uniform 1000 mul 500 sub % -500 <y<500
        25 background [0.270 0.035 0.058] /ink]
    } repeat
    100 {
      [random:uniform 1000 mul 500 sub % -500 <x<500
        random:uniform 1000 mul 500 sub % -500 <y<500
        20 background [0.866 0.353 0.000 ] /ink]
    } repeat
    [ 300 200 300 /vortex]
    [ 0 -300 300 /vortex]](6,12)
\end{pspicture}

```

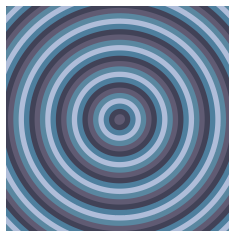
2.3 Concentric rings

Aubrey Jaffer describes the idea of “concentric rings” in:

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

“At the start of the real marbling process, inks are dropped from one or more locations to form expanding disks on a substrate. The mathematics is described in [Dropping Ink](#). For now, we just want an ink pattern which shows subsequent displacements. In my first renderings, 5 virtual inks are dropped from the center to form 25 concentric rings of equal radial width.

The boundaries between virtual ink rings will be traversed using the Minsky circle algorithm; although walking the circles using coordinates generated by sin and cos would work as well. The angular step size is made inversely proportional to the ring radius, making the distance between successive points uniform.”



```

\begin{pspicture}(-2,-2)(2,2)
\psMarble[actions={concentric-bands}](3,3)
\end{pspicture}

```

3 The command \psMarble

`\psMarble [Options] (x,y)`

The command `\psMarble` contains the options `actions=`, `background=`, `resolution=`, `viscosity=`, `L=`, `drawcontours=true/false` and `colors=`.

The optional argument `(x,y)` gives the x and y dimension of the image—the default is `(10,10)`.

Name	Default	Meaning
actions	concentric-bands	the type of marbling
background	0 0 0 [0.376 0.357 0.451] [0.255 0.259 0.341]	background color
colors	[0.306 0.498 0.62] [0.686 0.741 0.851] [0.353 0.529 0.627]	colors of the marbling within the rgb-system
drawcontours	false	if set to true it only draws the contours
resolution	1	resolution=1 is the pixel size 1 pt, resolution=0.5 is the pixel size 0.5 pt
viscosity	25	system (ink/fluid) constant
L	200	system (ink/fluid) constant (if taken negative, it pixels)

Notes:

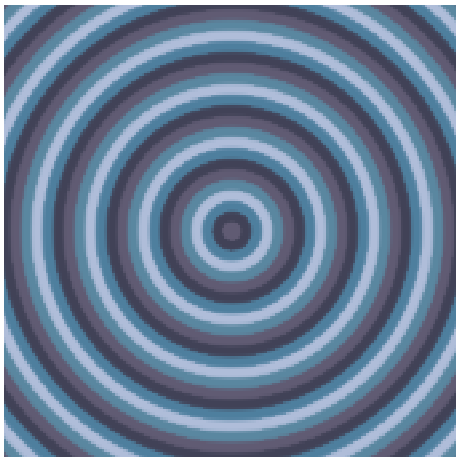
- If $L < 0$, the image will be pixeled. The size of the pixels can then be changed by the option `resolution=1` or `resolution=0.5`. If $L > 0$, the option `resolution=` is out of effect.
- Following are introduced some basic deformation actions, like `/line`, `/stroke`, `/circle`, `/vortex`, `/wiggle`.
Within `/circle` and `/vortex` there are defined some radii r . If $r > 0$ is chosen, the deformation is anticlockwise, if positive the deformation is clockwise. The angle of deformation is calculated by: radius/PC (PC = distance to center)
- The boolean option `drawcontours` is by default set to false. If set to true, only the contours are drawn within the image.
- All colors are to be setup within the rgb-color-system.

4 Rendering

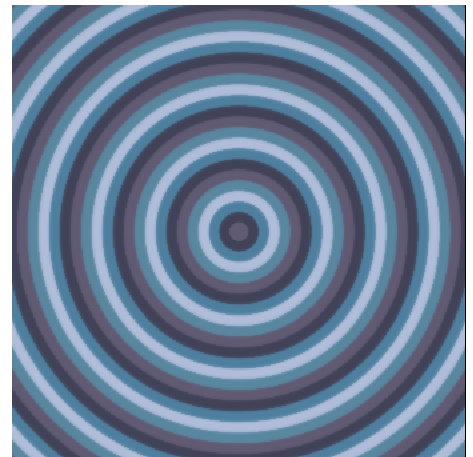
As designs get more complicated, hundreds of drops and strokes, reverse-rendering is the only practical way to render them. As the number of strokes increases, the number of points in the contours needs to increase as well. As the number of drops increases, the time to compute each pixel becomes less than the time to compute each contour-point on the drops.

The reason that we don't always reverse-render is because its resolution is limited to the raster; forward-rendering designs remain crisp at any magnification.

- To use forward-rendering (pixelfree) we choose the option $L > 0$ with a positive value.
- To use reverse-rendering (pixeled) we choose the option $L < 0$ with a negative value. In that case the option `resolution=` gets into action to vary the size of the pixels. Setting `resolution=1`, the pixel size is set to 1 pt; `resolution=0.5` sets the pixel size to 0.5 pt. In this case we consider the surface as a grid of points (square pixels of side 1pt) and each drop is represented by the points situated between its edges. When a new drop is placed, the points in that drop retain their color, the outer points are calculated before being assigned their initial color. This possibility is operational by taking for L (characteristic length of the system) a negative value.



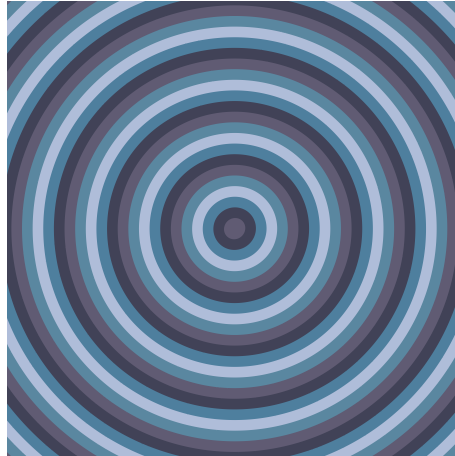
```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[L=-100,viscosity=20,
resolution=1,
actions={concentric-bands}](6,6)
\end{pspicture}
```



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[L=-100,viscosity=20,
resolution=0.5,
actions={concentric-bands}](6,6)
\end{pspicture}
```

5 Colors

All colors are setup within the rgb-color-system. Besides the preset colors= which are initially setup within the pst-marble.pro, we can change them within the concentric circles basic figure concentric-bands as follows:



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[actions={concentric-bands}](6,6)
\end{pspicture}
```



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[colors={
    [0.812 0.961 0.522]
    [0.745 0.937 0.357]
    [0.655 0.878 0.212]
    [0.541 0.780 0.071]
    [0.416 0.612 0.002]
  },
actions={concentric-bands}](6,6)
\end{pspicture}
```

6 Predefined objects before deformation

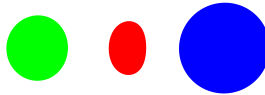
6.1 /ink

/ink defines a drop set on the surface of a liquid.

```
xc yc R      r g b
[ 0 0 100 [] [0 0 1] /ink]
```

This initially is a circle with the center (x_c, y_c) and radius R . The ink color is defined by the rgb-System. The empty slot `[]` is the background ink color.

In order to interpolate the color in reverse-rendering, the adjacent color must be known. Code in /marble initializes the empty slot from the inks in /actions.



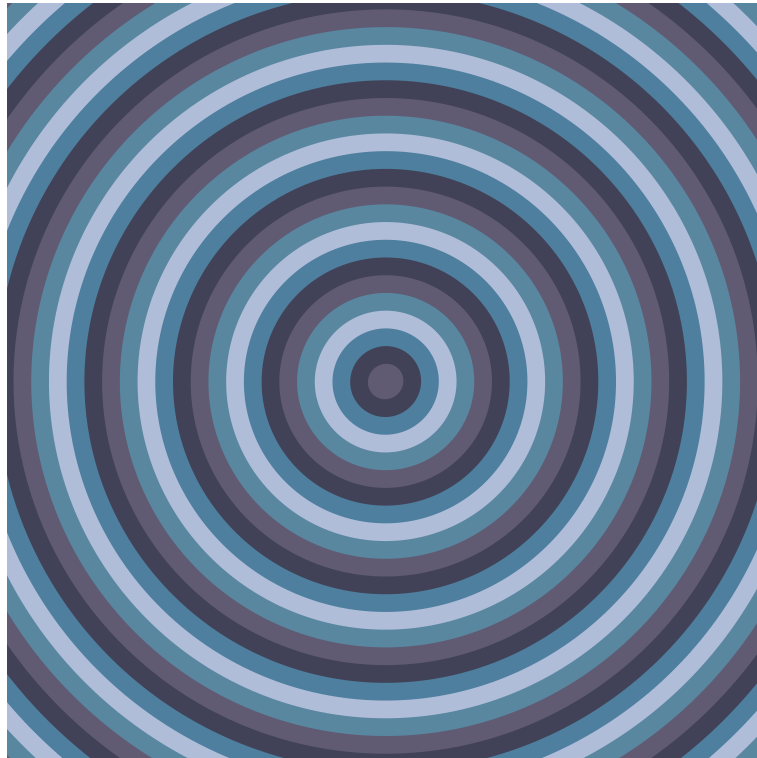
```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[background=1 1 1, %white
actions={%
[ 0 0 50 [] [1 0 0] /ink]
[ -200 0 70 [] [0 1 0] /ink]
[ 200 0 100 [] [0 0 1] /ink]
}](6,6)
\end{pspicture}
```

Note: The ink drop top most on the stack is left undeformed (intact), whereas all the others are influenced by each other according to the system constants. There are as well 6 colors predefined which can be used like that:

```
[ 0 0 50 [] color1 /ink]
[ -200 0 70 [] color2 /ink]
[ 200 0 100 [] color3 /ink]
```

6.2 concentric-bands

A set of concentric circles can be setup by `actions=concentric-bands` to then let some deformation happen.



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[actions={concentric-bands}](10,10)
\end{pspicture}
```

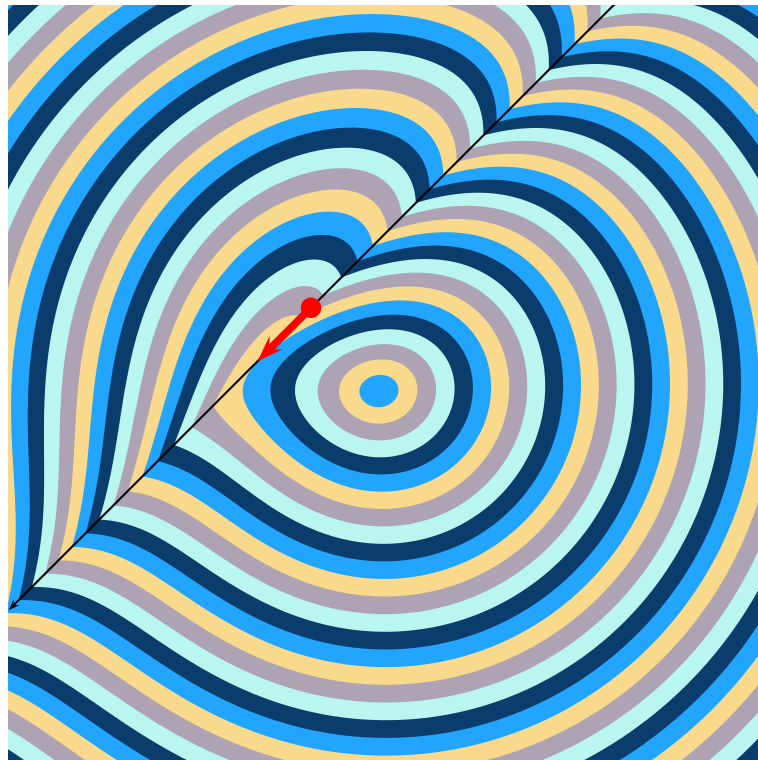

7 Basic deformation actions

Some of the deformation actions= which are initially setup within the pst-marble.pro can be manually changed by its parameters:

7.1 /line-deformation

The documentation is on the web page:

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>



```
\begin{pspicture*}(-5,-5)(5,5)
\psMarble[L=100,viscosity=50,
  colors={
    [0.134 0.647 1.0 ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942 ]
    [0.04 0.236 0.424 ]
  },
  actions={concentric-bands
% pour les formules voir :
% http://people.csail.mit.edu/jaffer/Marbling/Mathematics
% px py dx dy xs [ ys ] --> px py
% px py dx dy [ xs ] ys --> px py
% dx dy direction du déplacement
%   dx dy   xB   yB
[ -1 -1 normalize [ -100 ] 100 /line]
  }}]
```

```

\pstVerb{/scaleFactor 10 1000 div def
        /xB1 -100 scaleFactor mul def
        /yB1 100 scaleFactor mul def
        }%
\rput(!xB1 yB1){\psline{->}(5,5)(-4,-4)
        \psline[linecolor=red,linewidth=0.1]{->}(-1,-1)
        \psdot[linecolor=red,linewidth=0.1]}
\end{pspicture*}

```

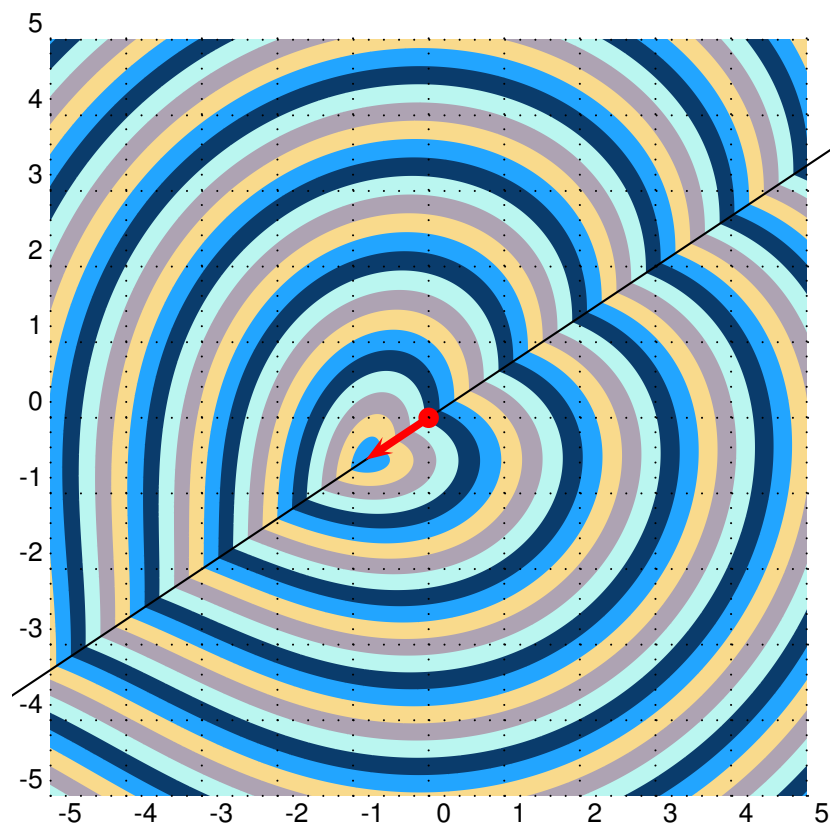
The base figure is setup by a family of concentric colored circles: “concentric-bands”, this command can be found within the file `pst-marble.pro` of the package. The command `/line` applies to all the points (x,y) of the circle.

```

x y dx dy xs [ ys ] --> x' y'
x y dx dy [ xs ] ys --> x' y'

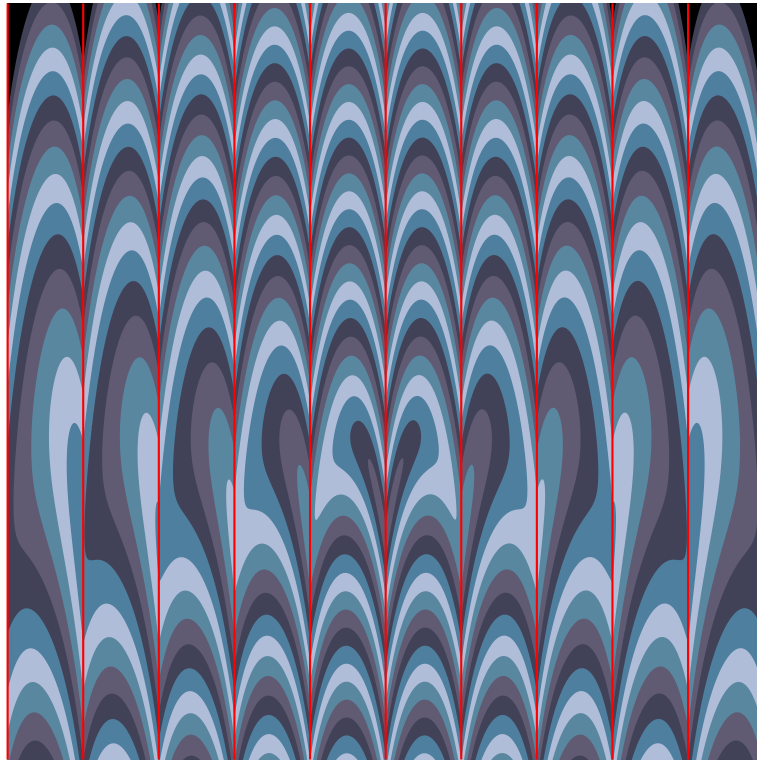
```

The line is determined by a point (x_S, y_S) and a unitary vector given by the couple of values (dx, dy) which determine the direction of the displacement. Within the command, the coordinates (x, y) are automatically put on the stack. The length and amplitude of the displacement are coupled to the viscosity and the characteristic system constant. This unitary vector (dx, dy) could if necessary be normalized by the following `-1.5 -1 normalize`. One now can setup different values for L and viscosity to observe the resulting effects.



```
[ -1.5 -1 normalize [ 0 ] 0 /line]
```

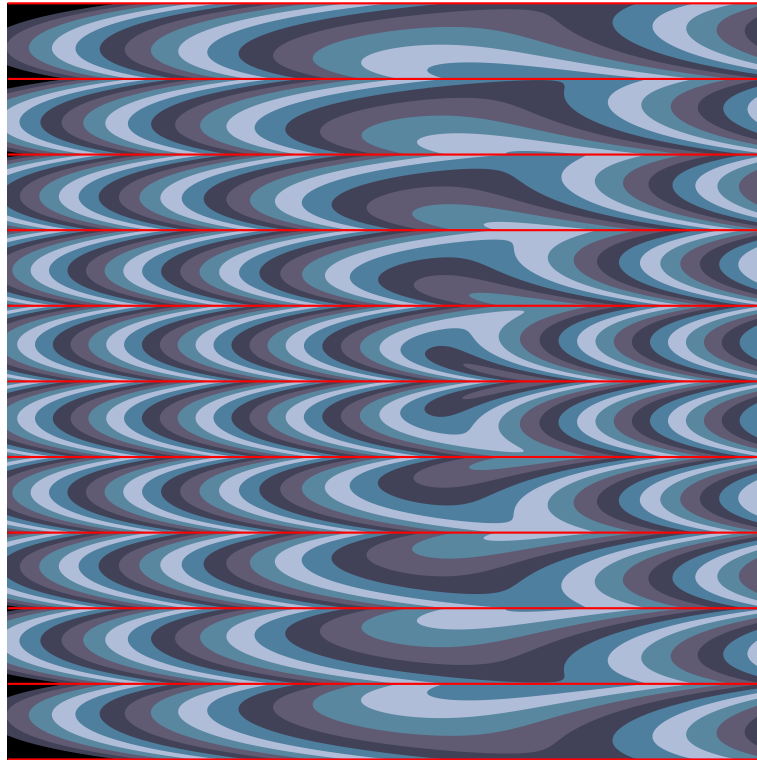
One can setup a series of values for x_S in brackets: `[xs1 xs2 ...]`, or for y_S : `[ys1 ys2 ...]` which define a series of deformation lines.



```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[L=200,viscosity=20,
  actions={concentric-bands
% pour les formules voir :
% http://people.csail.mit.edu/jaffer/Marbling/Mathematics
% px py dx dy xs [ ys ] --> px py
% px py dx dy [ xs ] ys --> px py
% dx dy direction du déplacement
%      dx dy      xs1  xs2                                ys
%   [ 0 -1 [ -500 -400 -300 -200 -100 0 100 200 300 400 500 ] 0 /line]
%       [ 0 -1 [ -500 100 500 {} for ] 0 /line]
%           }]
\multido{\i=-5+1}{11}{\psline[linecolor=red](\i,-5)(\i,5)}
\end{pspicture}

```



```

\psMarble[L=200,viscosity=20,
  actions={concentric-bands
% pour les formules voir :
% http://people.csail.mit.edu/jaffer/Marbling/Mathematics
% px py dx dy xs [ ys ] --> px py
% px py dx dy [ xs ] ys --> px py
% dx dy direction du déplacement
%      dx dy xs  ys1 ys2 .....
      [1 0  0 [ -500 100 500 {} for ] /line]
    }]
\multido{\i=-5+1}{11}{\psline[linecolor=red](-5,\i)(5,\i)}
\end{pspicture}

```

7.2 /stroke-deformation

The documentation is on the web page:

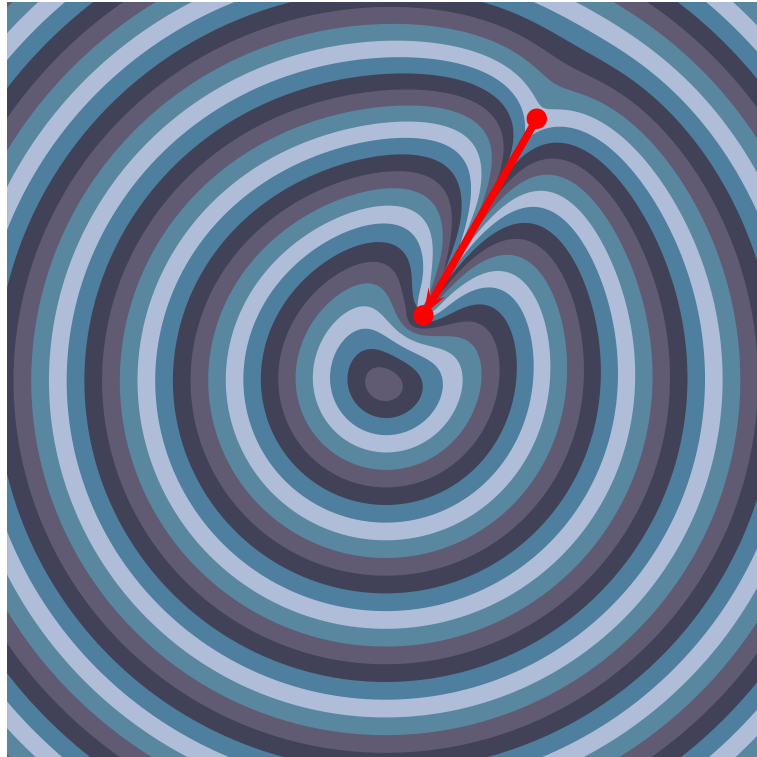
<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

Parameters for /stroke-deformation: bx, by, ex, ey, wid

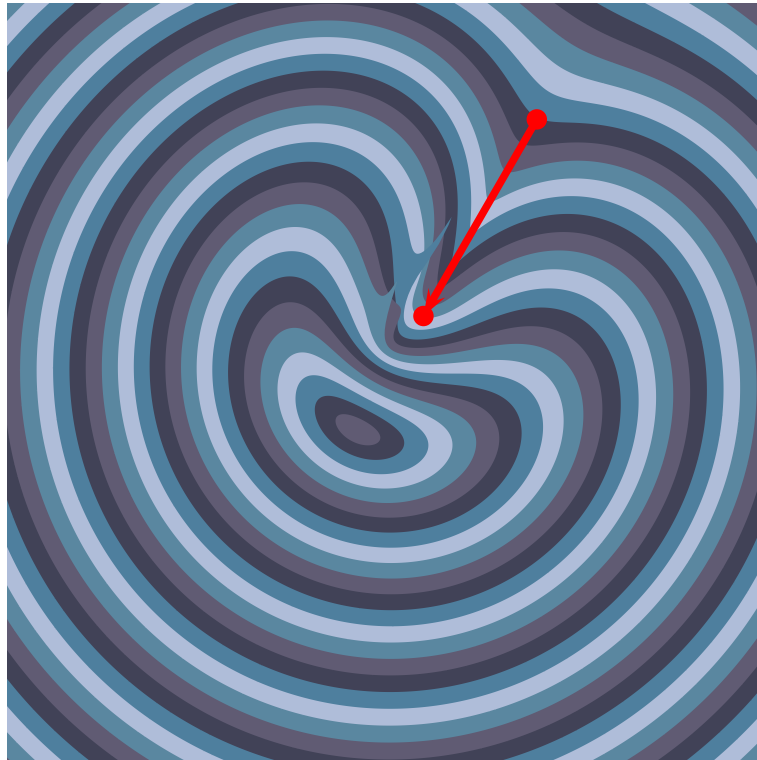
bx, by is the start point of the stroke. ex, ey is the end point of the stroke. wid is a system constant.

/stroke-deformation

% px py [bx by ex ey wid] --> px py



```
%Parameters: wid=50
\begin{pspicture}(-5,-5)(5,5)
\psMarble[L=200,viscosity=25,
actions={concentric-bands
% X1 begin Y1 begin X2 end Y2 end wid grandeur system
[ 30 sin 400 mul 30 cos 400 mul 30 sin 100 mul 30 cos 100 mul 50 /stroke ]
}](10,10)
\psset{linecolor=red,linewidth=0.1}
\pstVerb{/scaleFactor 10 1000 div def
/X1 30 sin 400 mul scaleFactor mul def
/Y1 30 cos 400 mul scaleFactor mul def
/X2 30 sin 100 mul scaleFactor mul def
/Y2 30 cos 100 mul scaleFactor mul def
}%
\psdots(!X1 Y1)(!X2 Y2)
\psline{->}(!X1 Y1)(!X2 Y2)
\end{pspicture}
```



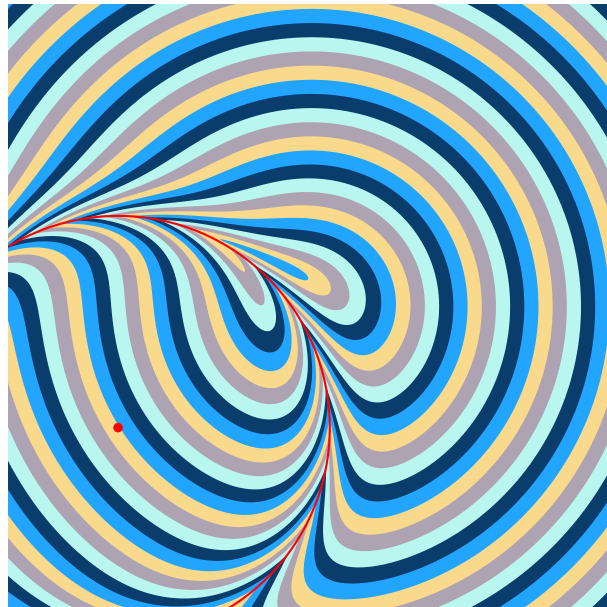
```
%Parameters: wid=200
\begin{pspicture}(-5,-5)(5,5)
\psMarble[L=200,viscosity=25,
actions={concentric-bands
% X1 begin Y1 begin X2 end Y2 end wid grandeur system
[ 30 sin 400 mul 30 cos 400 mul 30 sin 100 mul 30 cos 100 mul 200 /stroke ]
}](10,10)
\psset{linecolor=red,linewidth=0.1}
\pstVerb{/scaleFactor 10 1000 div def
/X1 30 sin 400 mul scaleFactor mul def
/Y1 30 cos 400 mul scaleFactor mul def
/X2 30 sin 100 mul scaleFactor mul def
/Y2 30 cos 100 mul scaleFactor mul def
}%
\psdots(!X1 Y1)(!X2 Y2)
\psline{->}(!X1 Y1)(!X2 Y2)
\end{pspicture}
```

7.3 /circle-deformation

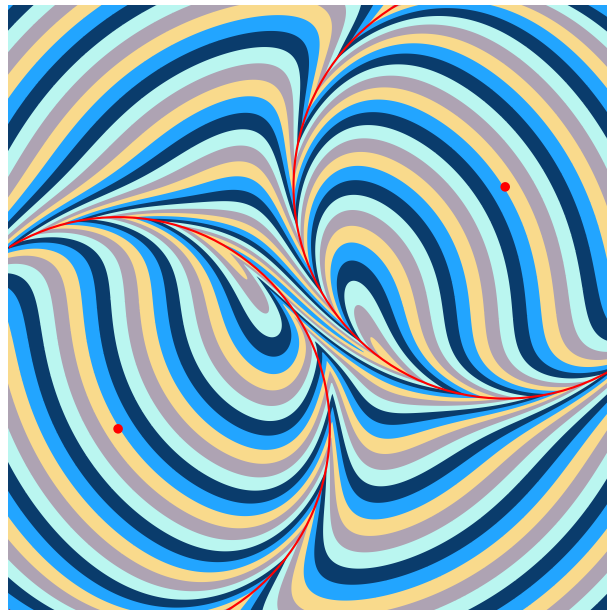
The parameters to be passed to the command `/circle` are the coordinates of the center of the circle (`xC yC`) and its radius `r` in pts. If `r` is set to positive values the circle deformation goes anti-clockwise otherwise clockwise.

```
[ xC yC [ r ] /circle ]
```

Within the brackets one can setup multiple radii, see example on page 33.



```
\begin{pspicture*}(-4,-4)(4,4)
\psMarble[
  colors={
    [0.134 0.647 1.0 ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942 ]
    [0.04 0.236 0.424 ]
  },
  actions={concentric-bands
% coordonnées et dimensions en pts
%          xC    yC    rayon
%          [ -320 -200 [ 350 ] /circle ]
  }](8,8)
% dimension de la page 8 cm
% le facteur d'échelle 8/1000
% les coordonnées du centre et le rayon sont à calculer avec ce facteur
\psset{linecolor=red}
\pstVerb{/scaleFactor 8 1000 div def}%
\psdot(!-320 scaleFactor mul -200 scaleFactor mul) % le centre
\pscicle(!-320 scaleFactor mul -200 scaleFactor mul){!350 scaleFactor mul}
\end{pspicture*}
```



```

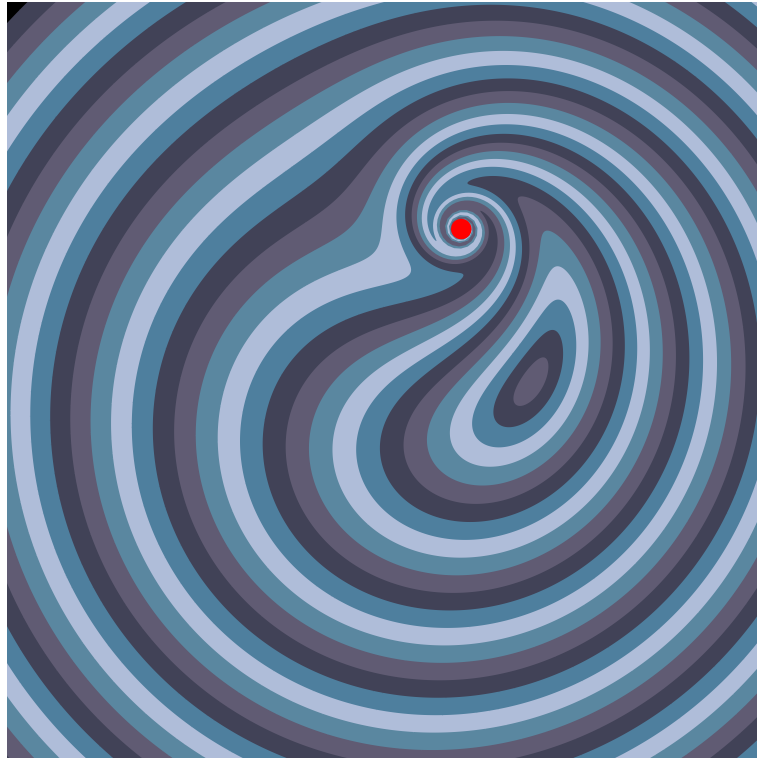
\begin{pspicture*}(-4,-4)(4,4)
\psMarble[
  colors={
    [0.134 0.647 1.0 ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942 ]
    [0.04 0.236 0.424 ]
  },
  actions={concentric-bands
% coordonnées et dimensions en pts
%
      xC    yC    rayon
      [ -320 -200 [ 350 ] /circle ]
      [  320  200 [ 350 ] /circle ]
    }](8,8)
% dimension de la page 8 cm
% le facteur d'échelle 8/1000
% les coordonnées du centre et le rayon sont à calculer avec ce facteur
\pstVerb{/scaleFactor 8 1000 div def}%
\psset{linecolor=red}
\psdot(!-320 scaleFactor mul -200 scaleFactor mul) % le centre
\pscircle(!-320 scaleFactor mul -200 scaleFactor mul){!350 scaleFactor mul}
\psdot(!320 scaleFactor mul 200 scaleFactor mul) % le centre
\pscircle(!320 scaleFactor mul 200 scaleFactor mul){!350 scaleFactor mul}
\end{pspicture*}

```

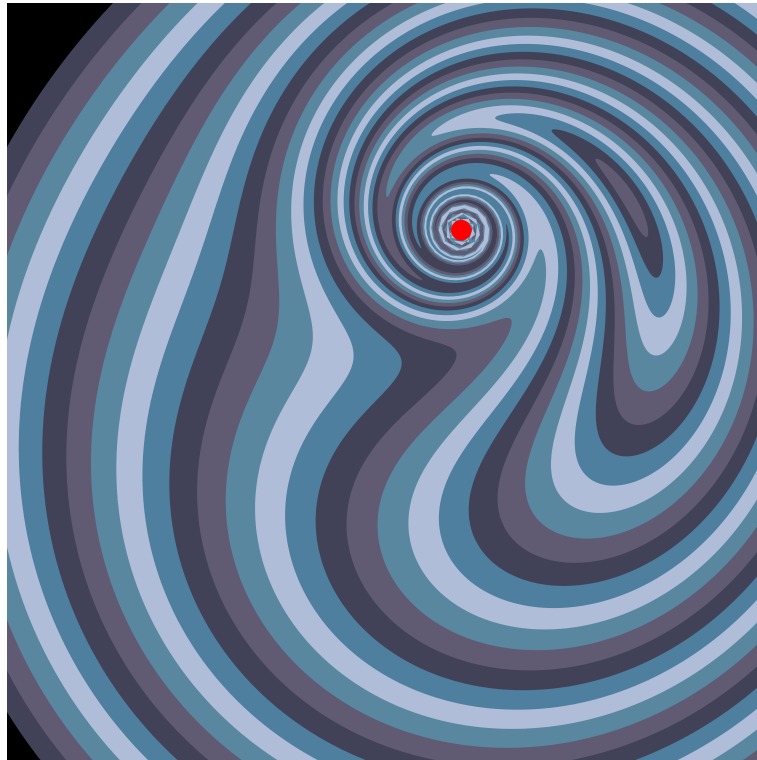

7.4 /vortex-deformation

Parameters for /vortex-deformation: cx, cy, rad

cx, cy is the center of the vortex. rad is the radius of the vortex. If rad is set to positive values the vortex deformation goes anti-clockwise otherwise it goes clockwise.



```
\begin{pspicture}(-5,-5)(5,5)
%Parameters: rad=200, L=200
\psMarble[L=200,
actions={concentric-bands
% cx cy rad
[100 200 200 /vortex]
}](10,10)
\pstVerb{/scaleFactor 10 1000 div def
/xC 100 scaleFactor mul def
/yC 200 scaleFactor mul def
}%
\psdot[linecolor=red,linewidth=0.1](xC yC)
\end{pspicture}
```



```

\begin{pspicture}(-5,-5)(5,5)
%Parameters: rad=500, L=200
\psMarble[L=200,
actions={concentric-bands
% cx cy rad
[100 200 500 /vortex]
}](10,10)
\pstVerb{/scaleFactor 10 1000 div def
/xC 100 scaleFactor mul def
/yC 200 scaleFactor mul def
}%
\psdot[linecolor=red,linewidth=0.1](xC yC)
\end{pspicture}

```

7.5 /wobble

The documentation is on the web page:

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

/wobble allows to realize deformations with an adequate function.

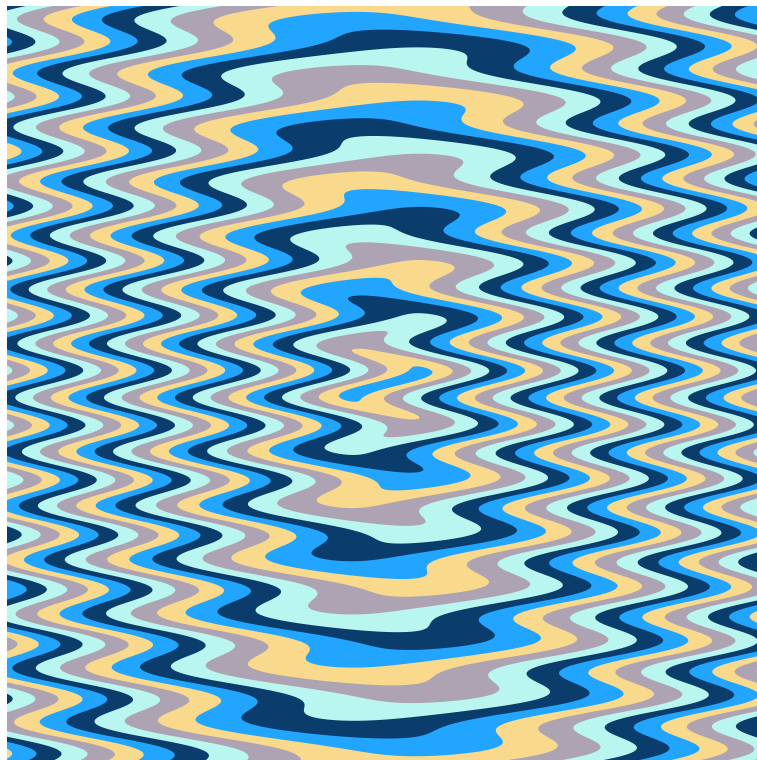
Aubrey Jaffer coded this following command:

```
/wobble % px py dx dy {func} --> px py
{
  /func exch def
  /dy exch def /dx exch def
  /py exch def /px exch def
  /a py dx mul px dy mul sub func exec reverse-rendering? {neg} if def
  px dx a mul add py dy a mul add
} bind def
```

(dx, dy) represents a unitary vector which gives direction to the wanted deformation by (u_x, u_y) .

$$a = f(yu_x - xu_y) \implies x' = x + au_x; y' = y + au_y$$

If one wants to obtain a sinusoidal wave parallel to the axis Oy , one takes $dx=1$ and $dy=0$ and for the function for example $45 \sin(5y)$ which in PS-code is: `{5 mul sin 45 mul}` and thus: $x' = x + 45 \sin(5y)$; $y' = y$. Note that the coordinates are calculated in points.



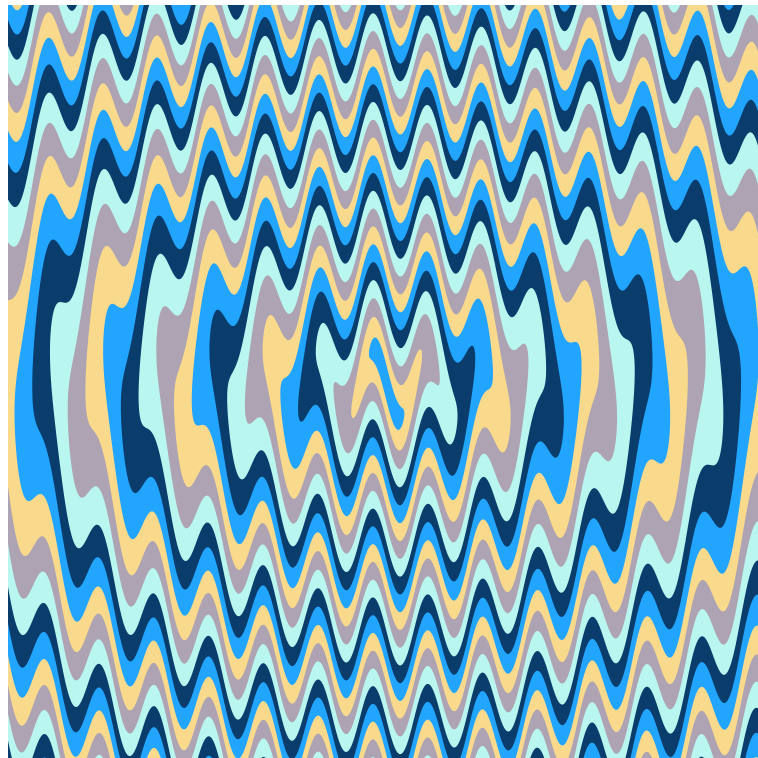
```
\begin{pspicture}(-5,-5)(5,5)
% sinusoïde d'axe // oy
\psMarble[L=100,
  colors={
    [0.134 0.647 1.0 ]
```

```

        [0.977 0.855 0.549]
        [0.684 0.638 0.702]
        [0.73 0.965 0.942 ]
        [0.04 0.236 0.424 ]
    },
    actions={concentric-bands
[1 0 {5 mul sin 45 mul} /wobble]
    }}
\end{pspicture}

```

To get waves that are upright, we write:

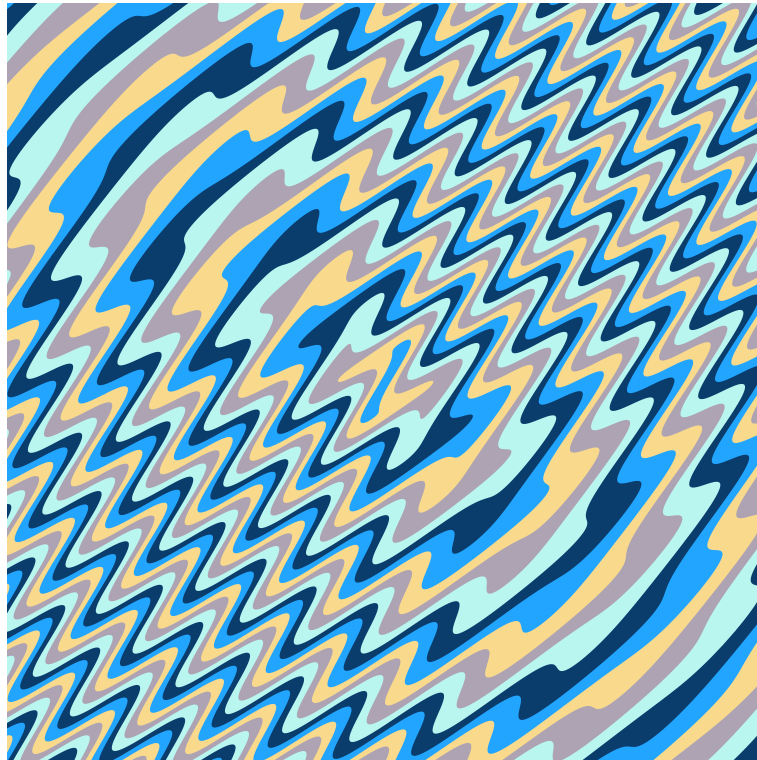


```

% sinusoïde d'axe // ox
\psMarble[L=100,
    colors={
        [0.134 0.647 1.0 ]
        [0.977 0.855 0.549]
        [0.684 0.638 0.702]
        [0.73 0.965 0.942 ]
        [0.04 0.236 0.424 ]
    },
    actions={concentric-bands
[0 1 {5 mul sin 45 mul} /wobble]
    }}
\end{pspicture}

```

For inclined/oblique waves, we write:



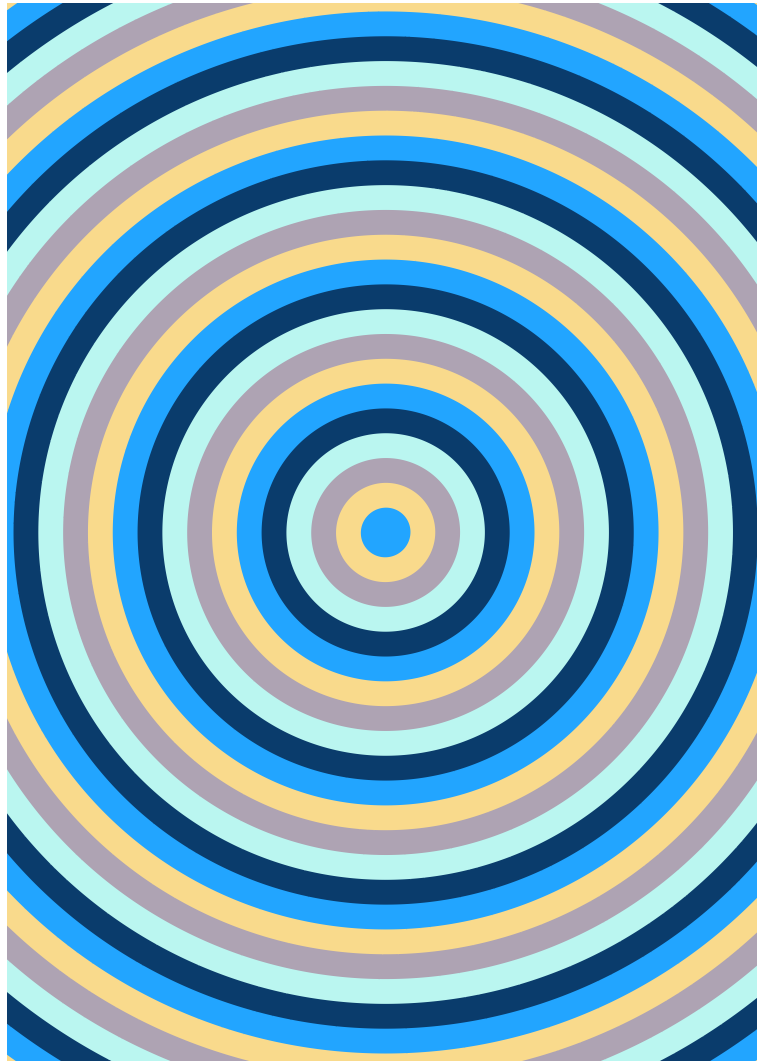
```

\begin{pspicture}(-5,-5)(5,5)
% sinusoïde oblique
\psMarble[L=100,
  colors={
    [0.134 0.647 1.0 ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942 ]
    [0.04 0.236 0.424 ]
  },
  actions={concentric-bands
    [1 1 normalize {5 mul sin 45 mul} /wobble]
  }]
\end{pspicture}

```

Note: If the vector is not unitary, we write the following to normalize it: 1 1 normalize.

Remark: These transformations can be combined and also be reversed, example:



```

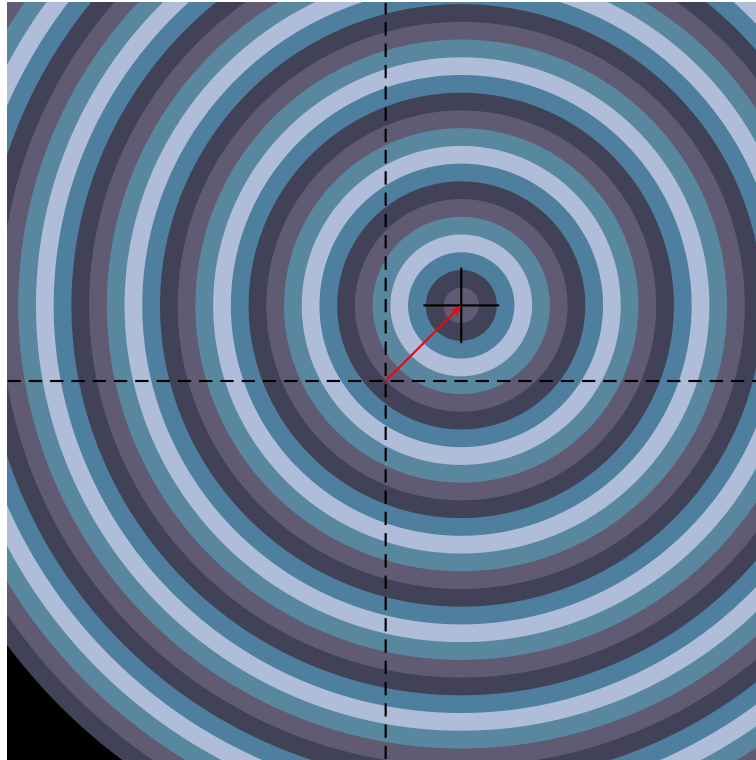
\begin{pspicture}(-5,-7)(5,7)
\psMarble[
  colors={
    [0.134 0.647 1.0 ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942 ]
    [0.04 0.236 0.424 ]
  },
  actions={concentric-bands
    [1 0 {3 mul sin 30 mul } /wobble ]
    [1 0 {3 mul sin -30 mul } /wobble ]
  }](10,14)
\end{pspicture}

```

7.6 /offset

```
/offset % px py dx dy --> px py
{
  /dy exch def /dx exch def
  dy reverse-rendering? {neg} if add exch
  dx reverse-rendering? {neg} if add exch
} bind def
```

The choice of units was made, so that 100 pts correspond to 1 cm within the image.



The displacement vector is given by dx and dy in pts.

```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[L=100,viscosity=50,
  actions={concentric-bands
%      dx dy
      [100 100 /offset]}}]
\pstVerb{/scaleFactor 10 1000 div def
  /xS1 100 scaleFactor mul def
  /yS1 100 scaleFactor mul def
}%
\rput(!xS1 yS1){\psline(0.5,0)(-0.5,0)\psline(0,0.5)(0,-0.5)}
\psline[linestyle=dashed](-5,0)(5,0)
\psline[linestyle=dashed](0,-5)(0,5)
\psline[linecolor=red]{->}(0,0)(1,1)
\end{pspicture}
```

8 Combined actions

Some basic actions set together. One can create some quite artistic images.

Note: The order of the actions is of importance! Also the direction of the deformation is important!

8.1 Bouquets



```
\begin{pspicture}(-4.5,-4.5)(4.5,4.5)
\psMarble[
  L=200,
  colors={
    [0.134 0.647 1.0 ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942 ]
    [0.04 0.236 0.424 ]
  },
  actions={
    concentric-bands
%    [ 0 0 20 background color1 /ink]
    [ 0 120 /offset]
    [ 0 -1 [ -460 60 460 {} for ] 0 /line]
    [ 1 0 {0.75 mul sin -120 mul} /wobble]
    [ 0 -1 [-750 -150 450] 0 /line ]
    [ 1 0 {0.75 mul sin 240 mul} /wobble ]
    [ 0 -1 [-450 150 750] 0 /line ]
    [ 1 0 {0.75 mul sin -120 mul} /wobble ]
  }](9,9)
\end{pspicture}
```


8.2 Fern



```
\begin{pspicture}(-5,-10)(5,10)
\psMarble[L=100,viscosity=20,
colors={
[0.659 0.859 0.573]
[0.467 0.718 0.357]
[0.306 0.573 0.192]
```

```

[0.180 0.431 0.071]
[0.086 0.286 0.000]
},
actions={concentric-bands
[ -1 0 0 [-400 50 400 {} for] /line]
[ 0 300 100 /vortex]
[ 0 100 100 /vortex]
[ 0 -100 100 /vortex]
[ 0 -300 100 /vortex]
}](10,20)
\end{pspicture}

```



```

\begin{pspicture}(-5,-5)(5,5)
\newsstyle{YellowGlass}{linewidth=0.1,linecolor=gray,
fillstyle=solid,fillcolor=gray}
\newsstyle{LensStyleHandle}{%
fillstyle=gradient,framearc=0.6,linewidth=0.5\pslinewidth,
gradmidpoint=0.5,gradangle=45,gradbegin=white,gradend=gray}
\psset{L=100,viscosity=10,background=1 1 1,
actions={
4 -1 1
{
/rad exch sqrt 100 mul def
[ 0 0 rad background color1 /ink]
[ 0 0 rad background color2 /ink]
[ 0 0 rad background color3 /ink]
[ 0 0 rad background color4 /ink]
[ 0 0 rad background color5 /ink]

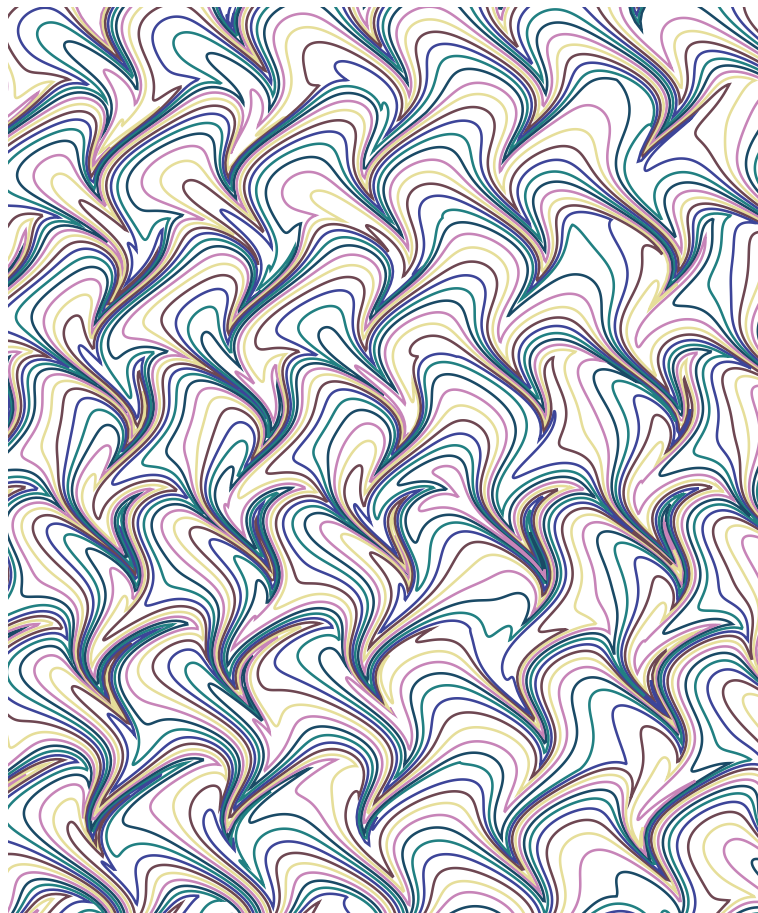
```

```

[ 0 0 rad background color6 /ink]
} for
[-800 0 [ -750 ] /circle] [800 0 [ -750 ] /circle]
[-500 0 [ -400 ] /circle] [500 0 [ -400 ] /circle]
[-300 0 [ -200 ] /circle] [300 0 [ -200 ] /circle]
[-250 0 [ -100 ] /circle] [250 0 [ -100 ] /circle]
}}%
\psMarble(10,10)
\PstLens[LensMagnification=2.5,LensRotation=45,LensSize=2,
        LensShadow=false,LensStyleGlass=YellowGlass](1,-1){%
\psMarble(10,10)}
\end{pspicture}

```

8.3 Draw contours only



```

\begin{pspicture}(-5,-6)(5,6)
\psMarble[L=25,viscosity=120,drawcontours,
        background=1 1 1,
        linewidth=0.1,
        actions={
        49 -1 1
        {
/cnt exch def
[ 0 0

```

```
700 50
div dup cnt 1 add mul dup mul exch cnt mul dup mul sub sqrt
[] colors cnt colors length mod get /ink ]
} for
[ 5 0 10 [ -600 150 600 {} for ] /line]
[ 0 -5 [ -600 150 600 {} for ] 0 /line]
[ 1 0 {2 mul sin 25 mul} /wobble ] },
colors={
    [0.9 0.87 0.6]
    [0.78 0.52 0.72]
    [0.1 0.29 0.4]
    [0.13 0.5 0.51]
    [0.24 0.27 0.6]
    [0.43 0.28 0.32]
}
](10,12)
\end{pspicture}
```

8.4 Ripple



```

\begin{pspicture}(-5.5,-8)(5.5,8)
\pstVerb{/ripple 75 def
/riplim ripple 6 mul def}%
\psMarble[L=500,viscosity=25,
  actions={
    11 -1 1
    {
/rad exch sqrt 40 mul def
[ 0 0 rad background color1 /ink]
[ 0 0 rad background color2 /ink]
[ 0 0 rad background color3 /ink]
[ 0 0 rad background color4 /ink]
[ 0 0 rad background color5 /ink]
[ 0 0 rad background color6 /ink]

```

```

    } for
    [ 0 50 /offset]
    [ 0 0 [ ripple ripple riplim {} for ] /circle ]
  }(11,16)
\end{pspicture}

```

8.5 Line-Line-Wiggle



```

\begin{pspicture*}(-5,-6)(5,6)
\psMarble[L=25,viscosity=120,
  actions={
    49 -1 1
    {
/cnt exch def
[ 0 0
700 50
div dup cnt 1 add mul dup mul exch cnt mul dup mul sub sqrt
[] colors cnt colors length mod get /ink ]
} for
[ 5 0 10 [ -600 150 600 {} for ] /line]
[ 0 -5 [ -600 150 600 {} for ] 0 /line]
[ 1 0 {2 mul sin 25 mul} /wiggle ] },
  colors={
    [0.9 0.87 0.6]

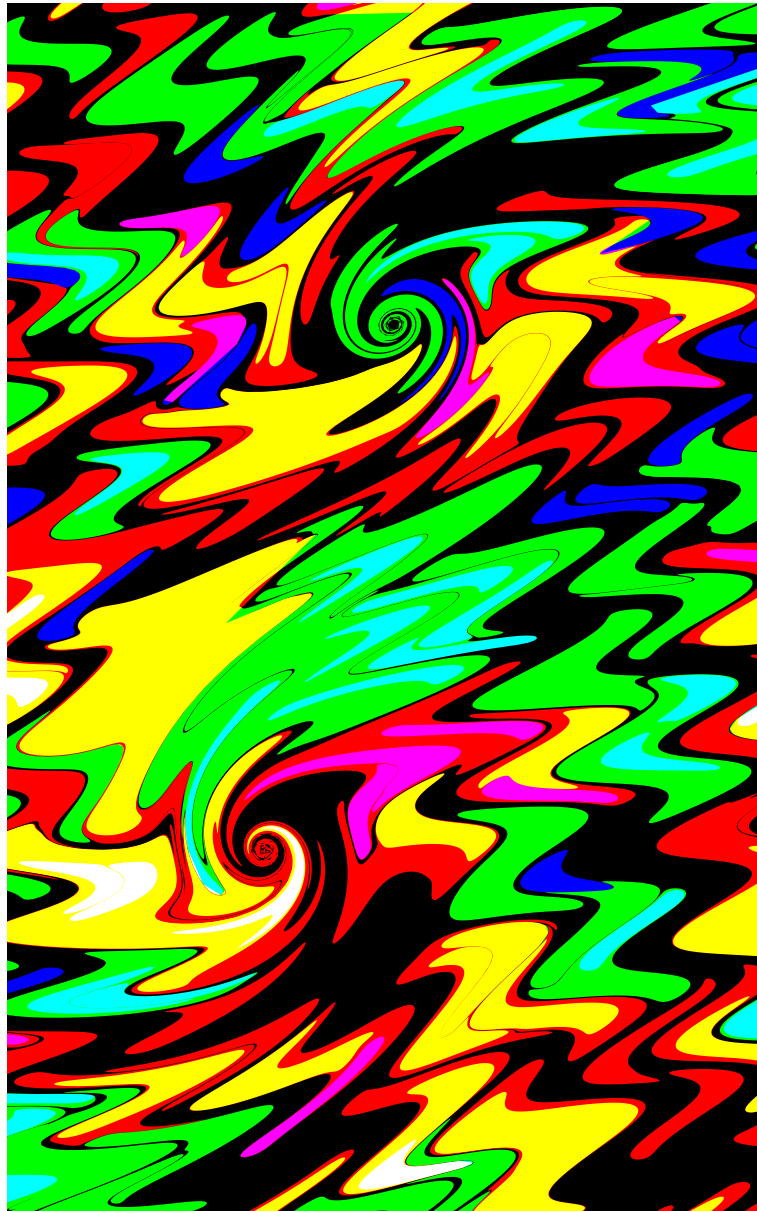
```

```

[0.78 0.52 0.72]
[0.1 0.29 0.4]
[0.13 0.5 0.51]
[0.24 0.27 0.6]
[0.43 0.28 0.32]
}
](10,12)
\end{pspicture*}

```

8.6 Blendmode



```

\begin{pspicture}(-5,-8)(5,8)
% /Lighten
% http://www.laboiteverte.fr/wp-content/uploads/2015/04/marbrure-papier-marbre-livre-16.jpg
\psMarble[L=100,
background=0 0 0,

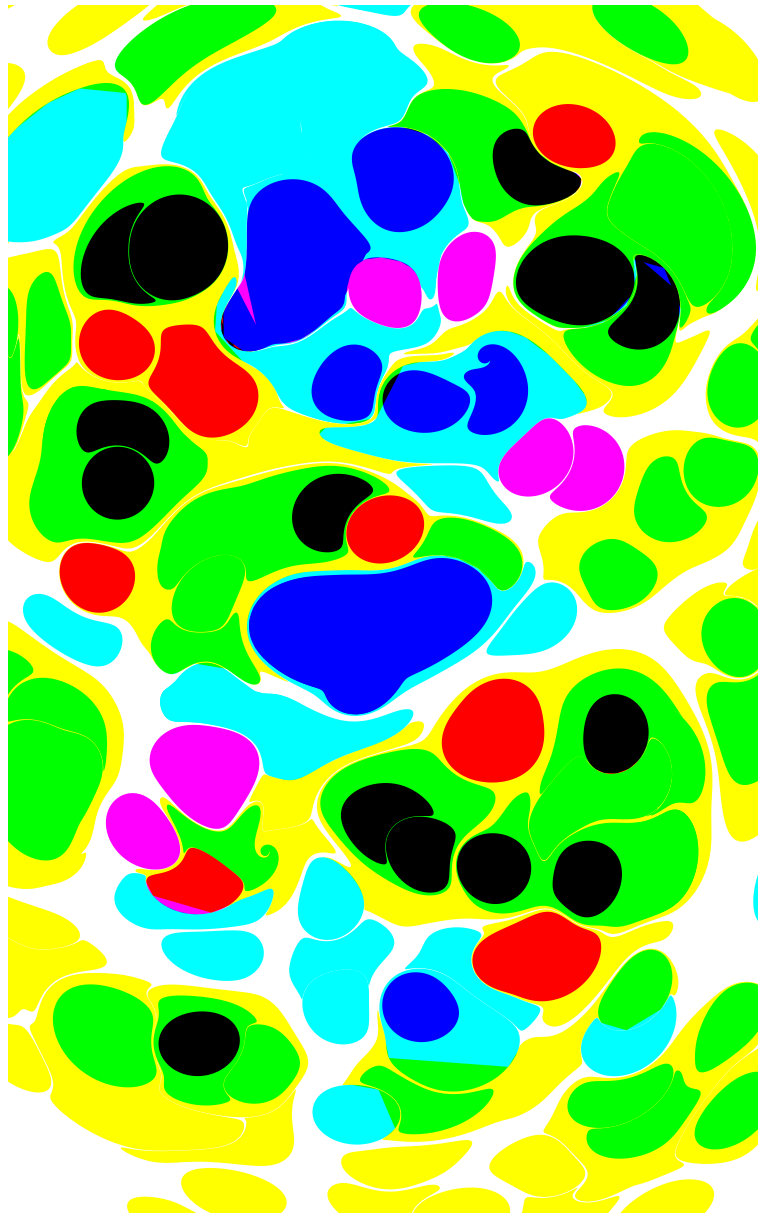
```



```

actions={
/Lighten .setblendmode
  200 {
[random:uniform 1000 mul 500 sub % -500 <x<500
 random:uniform 1000 mul 500 sub % -500 <y<500
 25 background [1 0 0] /ink]
} repeat
/Lighten .setblendmode
200 {
[random:uniform 1000 mul 500 sub % x
 random:uniform 1000 mul 500 sub % y
 30 background [0 1 0] /ink]
} repeat
/Lighten .setblendmode
100 {
[random:uniform 1000 mul 500 sub % x
 random:uniform 1000 mul 500 sub % y
 20 background [0 0 1] /ink]
} repeat
[1 0 {5 mul sin 45 mul} /wobble]
[ 100 200 100 /vortex]
[ -100 -200 100 /vortex}}](10,16)
\end{pspicture}

```

```

\begin{pspicture}(-5,-8)(5,8)
% /Darken
% http://www.laboiteverte.fr/wp-content/uploads/2015/04/marbrure-papier-marbre-livre-16.jpg
\psMarble[
  background=1 1 1,
  actions={
/Darken .setblendmode
    100 {
      [random:uniform pagewidth mul 1.5 mul pagewidth -2 div 1.5 mul round add    % x
        random:uniform pageheight mul 1.5 mul pageheight -2 div 1.5 mul round add % y
      30 background [1 1 0] /ink]
    } repeat
/Darken .setblendmode
    100 {
      [random:uniform pagewidth mul 1.5 mul pagewidth -2 div 1.5 mul round add    % x

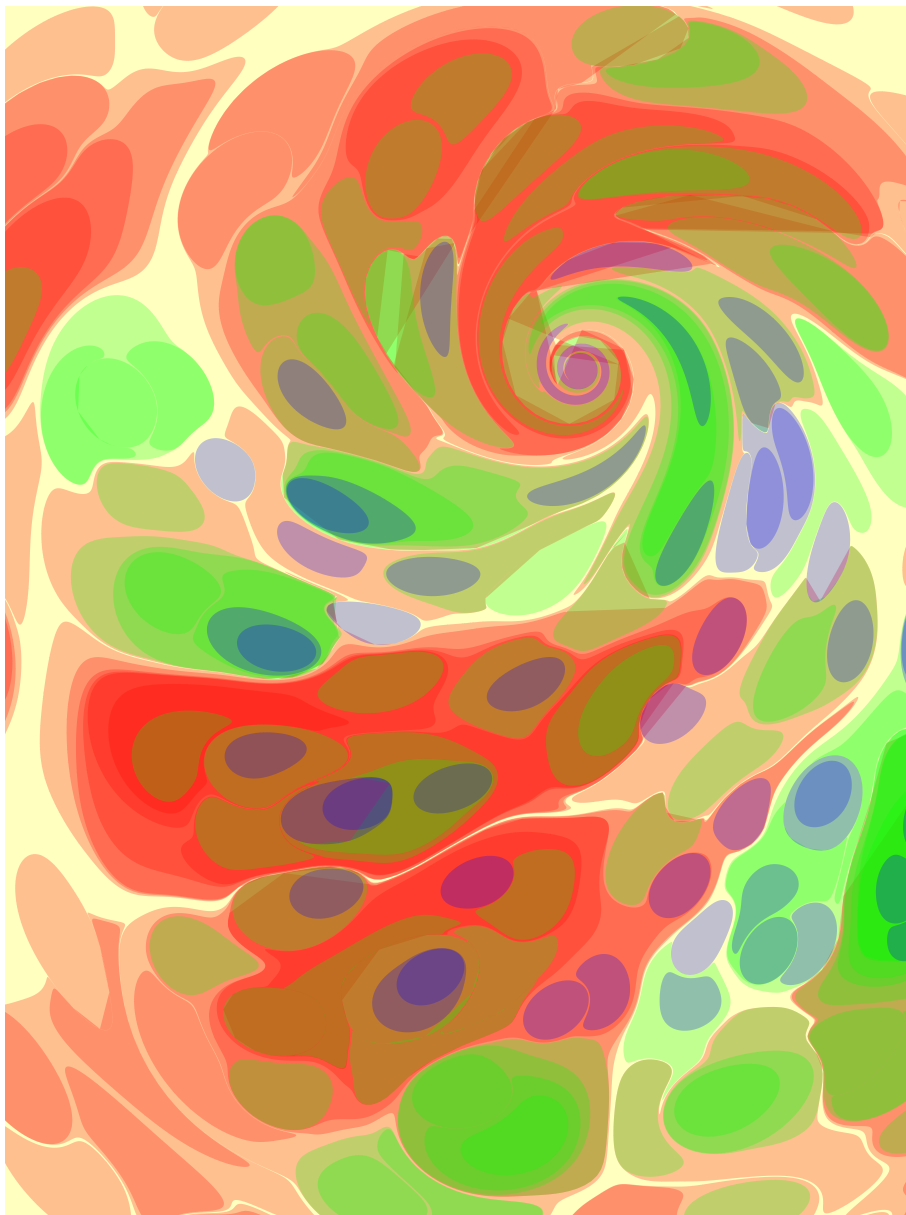
```

```

    random:uniform pageheight mul 1.5 mul pageheight -2 div 1.5 mul round add % y
    30 background[0 1 1] /ink]
  } repeat
/Darken .setblendmode
50 {
  [random:uniform pagewidth mul 1.5 mul pagewidth -2 div 1.5 mul round add % x
   random:uniform pageheight mul 1.5 mul pageheight -2 div 1.5 mul round add % y
   30 background [1 0 1] /ink]
  } repeat
  [ 100 200 20 /vortex]
  [ -100 -200 20 /vortex}]](10,16)
\end{pspicture}

```

8.7 Transparency



```
\begin{pspicture}(-6,-8)(6,8)
```

```

\psMarble[
  background=1 1 0,
  actions={
    0.25 .setopacityalpha
  100 {
    [random:uniform pagewidth mul 1.5 mul pagewidth -2 div 1.5 mul round add    % x
     random:uniform pageheight mul 1.5 mul pageheight -2 div 1.5 mul round add  % y
    45 [] [1 0 0] /ink]
  } repeat
    0.25 .setopacityalpha
  100 {
    [random:uniform pagewidth mul 1.5 mul pagewidth -2 div 1.5 mul round add    % x
     random:uniform pageheight mul 1.5 mul pageheight -2 div 1.5 mul round add  % y
    35 [] [0 1 0] /ink]
  } repeat
    0.25 .setopacityalpha
  50 {
    [random:uniform pagewidth mul 1.5 mul pagewidth -2 div 1.5 mul round add    % x
     random:uniform pageheight mul 1.5 mul pageheight -2 div 1.5 mul round add  % y
    25 [] [0 0 1 ] /ink]
  } repeat
  [100 200 200 /vortex]
  }](12,16)
\end{pspicture}

```

9 List of all optional arguments for *pst-marble*

Key	Type	Default
background	ordinary	[none]
viscosity	ordinary	[none]
L	ordinary	[none]
resolution	ordinary	[none]
actions	ordinary	[none]
colors	ordinary	[none]
drawcontours	boolean	true